BuildSense: Long-term, fine-grained building monitoring with minimal sensor infrastructure

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

Chayan Sarkar
TCS Research
Kolkata, India
sarkar.chayan@tcs.com

ABSTRACT

Buildings can achieve energy-efficiency by using solar passive design, energy-efficient structures and materials, or by optimizing their operational energy use. In each of these areas, efficiency can be improved if the physical properties of the building along with its dynamic behavior can be captured using low-cost embedded sensor devices. This opens up a new challenge of installing and maintaining the sensor devices for different types of buildings. In this article, we propose BuildSense, a sensing framework for fine-grained, long-term monitoring of buildings using a mix of physical and virtual sensors. It not only reduces the deployment and management cost of sensors but can also guarantee fine-grained, accurate data coverage for long-term use. We evaluate BuildSense using sensor measurements from two rammed-earth houses that were custom-designed for a challenging hot-arid climate such that almost no artificial heating or cooling is used. We demonstrate that BuildSense can significantly reduce the costs of permanent physical sensors whilst still achieve fit-for-purpose accuracy and stability.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Redundancy*;

KEYWORDS

energy-efficient building, sensor data estimation, virtual sensing, sensing as a service

ACM Reference format:
Rachel Cardell-Oliver and Chayan Sarkar. 2017. BuildSense: Long-term, fine-grained building monitoring with minimal sensor infrastructure. In *Proceedings of The 4th International Conference on Systems for Energy-Efficient Built Environments, Delft, The Netherlands, November 8-9, 2017 (BuildSys)*, 10 pages.
DOI: http://dx.doi.org/xx.xxxx/xxxxxxx.xxxxxxx

1 INTRODUCTION

Residential and commercial buildings account for almost 21% and 18% of total U.S. energy consumption, respectively [20]. This has a direct impact on greenhouse gas emission, thus on environmental health. Emission from the buildings can be reduced either by reducing emissions from the energy supply (e.g., energy generation through renewable sources) or by reducing energy consumption through improved building design and lower energy use [20]. A large body of work aims to make buildings more energy-efficient through better utilization of the HVAC [1, 10] and the lighting systems [8, 17]. These systems depend on continuous data collection by a large number of sensors and an established physical model of the buildings.

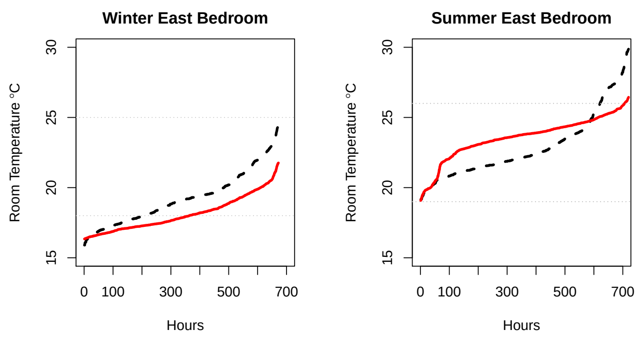

Figure 1: Measured (solid line) and modeled (dotted) indoor temperatures may not agree. Recommended set points for heating/cooling are in dashed gray in summer/winter.

On the other hand, new building designs and new building materials are also investigated to reduce energy consumption. However, there is a lack of scientific evidence about the long-term performance of new building designs in real-world settings [6]. For example, a long term study of the houses built from rammed-earth showed that the state-of-the-art building models significantly underestimated performance and subsequent comfort, and thus over-estimated energy use [3]. Figure 1 shows both measured and modeled temperature distributions in one bedroom of a rammed-earth house for one winter and one summer month. The horizontal gray lines indicate the recommended set points for applying artificial heating/cooling in summer/winter. If the room temperature goes above the upper set-point then cooling would be applied or if below the lower set-point then heating. It is clear from the figure that the distribution of modeled (dashed line) and measured (solid red line) temperatures differ markedly. That is, the performance of these buildings is not yet well represented in the building physics models. For energy rating purposes, a tally is made of the hours *outside* a comfortable temperature range. Since the modeled and actual hours in these temperature ranges differ significantly, the predicted energy requirements of the houses will be inaccurate.

Problem description. With the advent of embedded technologies, automated data collection using embedded sensor devices has become mainstream. To understand an indoor space, a trivial solution would be to deploy sensors at every possible corner of the indoor space, collect data continuously, and infer the current conditions at different parts of the space. However, there are two major problems with this. Firstly, even if the sensor devices become cheaper, deploying them in large numbers still incurs a significant cost for purchasing, deployment, and management. Secondly, for certain locations the permanent installation of physical sensors is infeasible. For example, head-level temperature sensors would be inconvenient for residents. Though using energy harvesting and parsimonious scheduling reduces the maintenance cost of changing batteries, it neither solves the problem of long-term deployment of large numbers of sensors nor the problems of tackling sensor failures or that some sensor positions are not acceptable for building residents. Thus, a different approach is required that can provide a fine-grained, continuous sensor measurement without the cost and management burden of a large permanent sensor deployment.

Approach. In this article, we aim to reduce the costs of sensor management by mixing real sensors with virtual sensors, where virtual sensors are models that can use the readings of one sensor to predict sensed data at another location even if there is no physical sensor present. The obvious question is how to create the virtual sensors and how reliable are they? Our approach comprises a short training period and a long term operational period. During the training period, temporary sensors are deployed in the building and measurements are collected. This data is used to train virtual sensors that predict the readings of a target sensor using the readings of the predictor sensor [5, 25]. The selection of predictor sensors is data driven in that the predictor sensors are not necessarily near the target sensor and may not even sense the same phenomena.

After the training period, the whole system is analyzed to determine an optimal mix of physical and virtual sensors to achieve the purpose of the sensor network. Sensor selection is a multi-objective optimization problem balancing the accuracy and robustness of the predictions, with the cost of deployment. During the operational period, the selected deployed and virtual sensors are used to deliver data for the application.

The main problem addressed in this article is how to select an optimal set of physical and virtual sensors for long term monitoring so that the number (or other costs) of deployed sensors is minimized and the reported values of the sensor network are within an acceptable error bound for the domain application. This is particularly important for companies who are selling **sensing as a service**. They do not charge their clients based on the number of devices, rather for the knowledge gained from the gathered data. If they can meet the application requirement with minimal fixed hardware that increases their revenue. The main contributions are summarized in the following.

- We propose a new algorithm for measuring a 3D space using an optimal mix of physical and virtual sensors for a building sensor network.
- We evaluate the algorithm using real world building networks, demonstrating that there are opportunities for significantly reducing the complexity of sensor network deployments and the ongoing costs of their maintenance.
- We show that our algorithm is able to maintain accurate predictions over a long time period of more than a year using only one month of training data.

2 BACKGROUND

A large body of prevalent works focuses on how to achieve energy-efficient buildings. There are two fundamental approaches – (i) reducing electricity consumption, and (ii) constructing energy-efficient building structures. In this section, we discuss the pivotal role of sensor networks for both of these categories. Since managing a sensor network is not a trivial task, a number of approaches are used to enable a sensor network as a simple data collection tool. We briefly discuss some of these approaches along with their limitations as a building monitoring tool.

2.1 Electricity consumption reduction

HVAC and lighting systems are identified as the highest contributors to the energy consumption in a building [20]. As a significant proportion of this energy consumption is attributed to inefficient usage, there is scope for significant energy saving by using efficient means to manage HVAC and lighting units. One of the prominent approaches is to identify the occupancy of a room and control the devices accordingly [2, 10].

Though most works focus on identifying occupancy without deploying a large number of sensors, doing away with any sensors is not feasible. Another approach is to reduce the over-utilization of HVAC and lighting systems. This method requires a thorough understanding of how much usage is sufficient. Personal thermal comfort deals with this question [12, 21]. Similarly, lighting controls based on preference have also been studied [11, 26]. However, this can only be achieved if sufficient data is available to learn personal comfort and preferences, and ample data is available to assess the indoor conditions continuously.

2.2 Energy-efficient buildings

Effective control of energy in legacy buildings can certainly save energy. But when constructing new buildings the opportunity exists to use novel materials and designs to minimize the energy that will be required for occupants' comfort [3]. When a new building is planned, its performance is predicted at the design stage using building physics models, often supported by simulation software. Furthermore, many countries mandate energy efficiency standards for new buildings, requiring the new designs to be rated for their energy efficiency using these models [6]. However, existing models may not be accurate for novel designs. Thus, in-situ measurements are important for understanding the performance of novel designs, and ultimately for enabling effective policy on energy-efficient buildings.

2.3 WSN as a data collection tool

As sensors are an integral part of energy-efficient buildings, the goal of this article is to develop a building monitoring sensor system, which is economical, easily manageable, and provides accurate, fine-grained continuous data. Since the inception of wireless sensor networks (WSN), a large body of works has focused on fine-grained sensing from minimal measurements. Even though the primary

Table 1: Techniques to infer fine-grained sensor data from limited measurements.

category	general technique	limitations
data estimation	correlation in sensor data is exploited to predict one sensor value with the help of other	applicable only for short-term estimation
coverage problem	avoid overlapped-sensing by multiple sensors; use only a subset of sensors that can cover the whole area (or all the points)	*a priori* knowledge about the field is required and the statistics should hold true over time
node scheduling	select only a subset of sensors as active such that the whole monitoring region is covered avoiding overlap by multiple sensors	scheduling can be done only in short burst to cope with varying dynamics of the field
compressed sensing	exploiting sparsity in the sensed data, recover the whole dataset from very few samples	*a priori* knowledge about the field is required and the data should be sparse in certain domain

objectives of these techniques overlap with each other, i.e., energy-efficiency and efficient sensor management, they differ in terms of their approach and application scenario. In the following, we briefly discuss these techniques by grouping them into four broad categories. A summary of these techniques is also provided in Table 1.

Data estimation: As data transmission consumes the most energy for an embedded sensor device, data estimation techniques are employed to reduce the amount of data transmission, thus energy consumption [9, 14, 23, 24]. They are also used to tackle missing sensor data, node failure, communication failure, etc.

Coverage problem: In many wireless sensor networks, there are more nodes than the optimal requirement. The reasons can be non-overlapping of the sensing range and the transmission range, infeasibility of careful deployment, tackling node failure, etc. This leads to redundant sensor nodes in the network. The coverage problem selects a subset of active nodes (post deployment) that are sufficient to cover the whole area while ensuring connectivity [4, 19]. The subset of active nodes are changed periodically such that there is balanced energy expenditure by the nodes.

Node scheduling: Node scheduling is similar to the coverage problem, where the data from a subset of deployed nodes are sufficient. However, here the active node selection is not only based on coverage criteria, but it can be based on spatial and/or temporal correlation among the nodes [15, 25, 28].

Compressed sensing: This is another technique that reduces the amount of traffic within the network. If the sensor data is sparse in some domain then using the compressed sensing technique sensor data of all the sources can be reconstructed while the measurement is done from a few sources [18, 27, 29].

2.4 Research gap analysis

It is evident that sensor-based data collection is an essential requirement for buildings to achieve energy-efficiency. However, most of the existing sensor network optimization techniques are applied at the post-deployment stage to minimize the use of deployed sensors. Though there are pre-deployment strategies for selecting a limited number of sensors [16, 22], they are most suitable for outdoor scenarios where statistics of the field is known *a priori*. In this work, we focus on a pre-deployment strategy for monitoring in buildings where the statistics of the field is not well known. We develop a method for fine-grained continuous data collection using only a minimal set of sensors (post-deployment).

3 BUILDSENSE FRAMEWORK

In this work, we present BuildSense, a framework for monitoring indoor spaces. The objective of the framework is to create and manage a sensor network for supporting energy-efficient buildings. Key features of BuildSense include optimal deployment strategy, low-cost and fault-tolerant deployment, ease of management, building-level customization, fine-grained and continuous data collection, robust and accurate sensor data. To attain all these features simultaneously, a number of design choices are made and accordingly algorithms are developed.

3.1 Design principles

At the core of its design, BuildSense needs to ensure that accurate, fine-grained, continuous sensor data is available. Additionally, its goal is to reduce the deployment and management cost of a large number of sensor devices. To fulfill these two divergent requirements, BuildSense uses an optimal mix of physical and virtual sensors, where virtual sensors use a prediction model that can predict sensor data accurately with the help of other physical sensors. The number of physical sensors is kept as low as possible, which leads to reduced costs. In this way, the use of virtual sensors ensures fine-grain measurement with a minimal sensor infrastructure. However, solving this problem raises a number of rudimentary questions – (a) how to create a virtual sensor? (b) how many physical sensors are required to ensure sufficient granularity in the data? (c) how to ensure the accuracy of the reported data over a long time period of months or years? BuildSense follows a sense-learn-predict model that can tackle these questions.

3.2 System Model

The BuildSense framework does not assume any special system model and can be integrated into any sensor network. In other words, it inherently supports easy customization for any building. The framework works in three phases – (i) data gathering, (ii) training, and (iii) operation, which not only allows building-level customization but also helps to tackle the rudimentary questions mentioned before. Figure 2 shows how the three phases constitute the framework.

During the **gathering phase**, a large number of sensors are deployed in the building. This temporary network collects data for a period of a few weeks or months where all the sensors report their values to a central node, called the sink node. This dense sensor deployment during the data gathering phase helps to understand

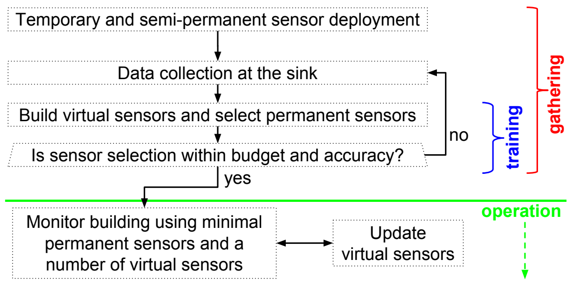
Temporary and semi-permanent sensor deployment
Data collection at the sink
Build virtual sensors and select permanent sensors
Is sensor selection within budget and accuracy?
no
yes
training
gathering
operation
Monitor building using minimal permanent sensors and a number of virtual sensors
Update virtual sensors

Figure 2: The three phases of the BuildSense framework.

the characteristics of the building and helps to ensure long term sensor data accuracy using minimal infrastructure. As mentioned earlier, the goal of BuildSense is to keep the number of permanent sensors as low as possible without violating the other constraints. As a result, a large number of sensors are removed after the data gathering phase and only a few remaining sensors become the permanent infrastructure.

The **training phase** commences either after sufficient data has been gathered or the data gathering and training phases can overlap (Figure 2). Various learning operations are performed during the training phase by utilizing the gathered data (details are in Section 4). When the learning functions attain a predefined level of confidence, the training phase, as well as the data gathering phase, are finished. Three tasks are performed during the training phase. **T1**. The first task is to learn prediction models for the virtual sensors in which the readings of a physical sensor are used to predict the readings of a virtual sensor. Sensor models are learned offline. **T2**. The second task is to select the best sensor types and positions for long term measurement, to achieve high prediction accuracy at a low infrastructure cost. **T3**. The third task is to adjust the selection of physical sensors to optimize fault-tolerance and the long-term stability of predictions.

Once the training phase finishes, the **operational phase** kicks in. The temporary sensors are removed along with some of the semi-permanent sensors; the rest become the permanent physical sensors. Selection of permanent sensors from amongst the semi-permanent sensors is discussed in detail in the next section. During operation, measurement commences using the physical (permanent) sensors and virtual sensor models are used to predict virtual sensor values.

4 LEARNING IN BUILDSENSE

BuildSense has three learning steps corresponding to each of the tasks. This section describes the algorithms for each of the learning steps as mentioned in Section 3. Before jumping into the algorithms, let us formalize the problem statement.

4.1 Problem Statement

Suppose, we have a set of sensed phenomena $S = \{s_1, \ldots, s_n\}$, each a time-series of observations so that $s_i(t) \in \mathbb{R}$ for timestamp t. Each sensor has a cost c_i reflecting its purchase, installation, and maintenance costs. More simply, costs can be binary for each sensor, i.e., whether to deploy permanently or not. There is an upper bound, *cost_max*, on the total allowed cost for a system.

If a sensor s_i can be estimated during a time period T by a predictor function $p_{ij}(s_j)$ within some error bound ϵ, so that $\forall t \in T,\ s_i(t) = p_{ij}(s_j(t)) \pm \epsilon$ then we say that s_i and s_j are ϵ-neighbors or p_{ij} has accuracy ϵ over time period T, written $s_i \approx s_j$ on T.

We use $P \subseteq S$ to refer to a set of permanent sensors and $V = S \setminus P$ the virtual sensors. Special permanent sensors $W \subseteq P$, called sentinels, are used to assess the stability of predictions. Stability is the percentage of time periods (e.g., months) during long term operation (e.g., years) for which predictions are sufficiently accurate. The lower bound (%) for stability is *min_stability*. For time period $Y = \{T_1, \ldots, T_M\}$ and sentinels $W \subseteq P$, we define the proportion of time periods for which predictions are stable by:

$$is\_stable(s_i, P, Y) = \Sigma_{k=1}^{M} (\exists\, s_j \in P \setminus s_i \mid s_i \approx s_j \ on\ T_k)\ /\ M.$$

Then, the problem we address is the following. Given training data from sensors $S = \{s_1, \ldots, s_n\}$, find a set $P \subseteq S$ of permanent sensors and $V = S \setminus P$ of virtual sensors and sentinels $W \subseteq P$ so that

(1) There is a sufficiently **accurate** prediction function for every virtual sensor based on a training period T, i.e.,

$$\forall s_i \in V,\ \exists\, s_j \in P.\ s_i \approx s_j \text{ on } T.$$

(2) The total **cost** of permanent sensors is bounded, i.e.,

$$\Sigma_{s_j \in P}\, c_j \ \leq\ cost\_max.$$

(3) Predictions are **stable** during long term operation of the network for all sentinel sensors in W, i.e.,

$$\forall s \in W,\ is\_stable(s, P, Y)\ \geq\ min\_stability.$$

4.2 Learning Virtual Sensors

The first task is to learn predictors for the virtual sensors. Algorithm 1 gives the pseudo code for this process. Given a field of sensors $S = \{s_1, \ldots, s_n\}$ that have gathered data during a time period T, the goal is to learn predictor functions for each pair of sensors s_i, s_j so that the predicted value of s_i using the current reading of s_j is within an error bound ϵ of the actual value ($s_i \approx s_j$).

Four predictor functions were investigated in detail in [5]: simple nearest neighbor, linear regression, cubic on the time of day [13], and nearest hour of day neighbor. Of these, the nearest hour of day predictor had the best performance. The nearest hour of day predictor uses a time of day adjustment for each hour of the day. An offset o_h for each hour of day $h \in \{0, 1, \ldots, 23\}$ is defined by $o_h = mean\{s_i(t) - s_j(t) \mid t.h = h\}$ where $t.h$ is the hour of day at time t. The predictor function for s_i is given by $p_{ij}(s_i) =_{def} \lambda t.\ s_j(t) + o_{t.h}$ and we write $s_j + o$ for this function.

Figure 3 illustrates how the nearest hour of day prediction works. The black lines show the readings for two sensors, s_1 and s_2, during a training period: s_1 is in a bedroom and s_2 is embedded in the wall of a different room. The red line in the lower part of the figure shows the learned hour of day offset function o for this sensor pair. The offset function has an amplitude of 1.4 ° C and it minimizes the error: $s_1 - (s_2 + o)$. This simple additive estimator turns out to be surprisingly effective. For this example, which includes some strong context changes in the weather, the root mean squared error (RMSE) for $s_1 - (s_2 + o)$ is 1.15° C over one month. This is sufficiently accurate for applications such as energy rating or HVAC control.

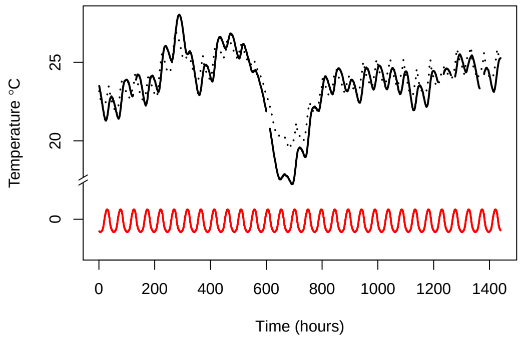

Figure 3: Sensor readings from a bedroom (dotted) and embedded in a wall (black) and their average difference by time of day (red).

Phenomena such as relative humidity can also be predicted, with RMSE of 2% to 4% [5].

Algorithm 1 generates hour of day offsets and RMSEs for every possible pair of sensors s_i and s_j. Lines 3 to 6 calculate the hour of day offset function, giving a virtual sensor estimator for s_i based on s_j. The error of this predictor is the RMSE of the residuals between the actual and estimated values for s_i (lines 7 and 8) using root-mean-squared error (RMSE) as the error statistic. The estimation errors are used in the sensor selection phase to optimize the selection of virtual sensors. Line 9 records the learned hour of day offset function for a sensor pair. Since this predictor is additive, the offset for sensor s_i using s_j is simply the negative of the offset for s_j using s_i.

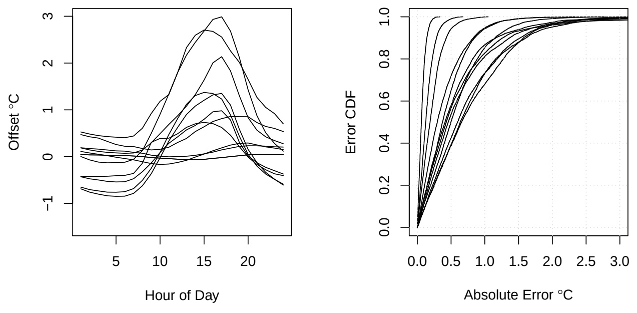

Figure 4: Offset vectors for nearest hour of day predictors (left) and CDF of the absolute error for these predictors (right).

Figure 4 (left) shows the set of offset vectors learned for 17 sensor pairs. It can be seen that offset values range from -1 to 3° C and that the diurnal shapes vary. The flexibility of the hour of day offset model allows more control than models such as nearest neighbor, or linear or cubic regression [5, 13]. CDFs of the absolute value of residuals are shown in Figure 4 (right). A few sensors pairs have 100% of their residuals below 1° C. In this sample, all virtual sensors have 90% of their observations within 1.5° C of the true value.

Algorithm 1: Learn a predictor for each sensor pair.

```
Input      : $S = \{s_1, \ldots, s_n\}$ set of sensors
             $T$ training period
Output     : $E$ matrix of prediction errors for each sensor pair
             $F$ prediction function parameters
Variables  : $p$ predictor sensor
             $d$ differences between sensor readings
             $o$ vector of learned hourly offsets
             $v$ virtual sensor to be predicted
             $r$ residuals between actual and predicted values
1  for $i \in 1 : n$ do
2      for $j \in (i+1) : n$ do
3          $p(T) = s_j(T)$
4          $d(T) = s_i(T) - s_j(T)$
5          $o = \langle h_0 \ldots h_{23} \rangle$ fitted from $d$
6          $v(T) = p(T) + o$
7          $r(T) = v(T) - s_i(T)$
8          $E[i,j] = RMSE(r(T));\ E[j,i] = E[i,j]$
9          $F(i,j) = o;\ F(j,i) = -1 \times o$
10 return $E, F$
```

4.3 Sensor Selection

During the sensor selection phase, two measures are considered – cost and accuracy. Algorithm 2 gives the pseudo code for the selection process. It chooses a set of permanent sensors $P \subset S$ so that all the virtual sensors $V = S \setminus P$ can be estimated with acceptable accuracy and total cost < *max_cost*. This is a version of the set cover problem, which is known to be an NP-hard problem [7]. However, there exists polynomial time greedy algorithm that works as follows.

Let P be a set of **predictor** sensors and C the pool of **candidate** sensors for a sensor network S. Initially $P = \emptyset$ and $C = S = U$ (line 1 of Algorithm 2). Or C can be restricted as will be explained shortly. For each iteration, we chose a sensor $s \in C$ that covers (i.e., able to predict) the highest number of yet to covered sensors (line 3). Add this chosen s to the set of permanent sensors and remove s and its ϵ-neighbors from the sets of candidate and uncovered sensors (lines 4 to 7). Finally, record the predictor function for each of permanent sensors' virtual neighbors (lines 8 to 9). Repeat the sensor selection steps until all candidate sensors are exhausted, i.e., $C = \emptyset$. Since every sensor can, at worst, predict itself, Algorithm 1 will always terminate. The algorithm returns P the set of permanent sensors to be deployed and the virtual sensors are $V = S \setminus P$ with predictors recorded in F_V.

The algorithm may fail to meet our requirements in two ways: the cost of permanent sensors may be too high, or it may not be possible to cover all sensors. First, we consider the cost failure case. Each sensor s_i has a usage cost c_i. This cost can take into account the purchase price, installation, and maintenance of that sensor. If all costs are the same for every sensor, then the algorithms

Algorithm 2: Select virtual and permanent sensors.

```
Input     : $S, E, F$ from Algorithm 1
            $PX \subset S$ (optional) temporary-only sensors
            $c : S \longrightarrow \mathbb{N}$ permanency cost for each sensor
            $\epsilon$ bound for predictor errors
            $cost\_max$ bound for permanent sensor costs
Output    : $V \subseteq S$ virtual sensors
            $P \subseteq S$ permanent, physical sensors
            $F_V \subseteq F$ chosen predictor functions
            $cost \in \mathbb{N}$ total cost of permanent sensors
Variables : $C$ candidate sensors for permanency
            $U$ undecided sensors
            $R$ virtual sensors predicted by the latest $s \in P$
1  $V = \emptyset; P = \emptyset; C = S \setminus PX; U = S$
2  while $C \neq \emptyset$ do
3      $s_i = max_{s_i \in C} \{size(\{s_j \mid E[i,j] < \epsilon\} \cap U)\}$
4      $P = P \cup \{s_i\}$
5      $R = \{ s_j \mid E[i,j] < \epsilon\} \cap U \}$
6      $U = U \setminus R$
7      $C = C \setminus R$
8      for $r \in R$ do
9          $F_V = F_V \cup \{F(r,s)\}\}$
10 if $U \neq \emptyset$ then
11     Warning: Cannot cover all sensors in $S$.
12 $cost = 0$;
13 for $s \in P$ do
14     $cost = cost + c(s)$
15 if $cost > cost\_max$ then
16     Warning: Cost of permanent sensors is too high.
17 $V = S \setminus P$;
18 return $P, V, F_V, cost$
```

simply minimize the number of permanent sensors. The sensor selection process takes account of high-cost sensors by allowing for an optional set $PX \subset S$, which is a set of temporary sensors that are too costly to deploy or maintain permanently. Such sensors are excluded from the candidates for permanency by initializing $C = S \setminus PX$ (line 1). Algorithm 1 returns the total cost of the deployment, that is the sum of the costs of sensors in P. It also gives a warning if the cost is too high (lines 15 to 16). The user can then increase ϵ or make more sensors available and rerun the algorithm to reduce the total cost. A topic for future work is to integrate cost considerations into the greedy selection of permanent sensors (line 3). This would require a utility function to balance the cost of a proposed permanent sensor against its coverage of virtual sensors. The task of evaluating different utility functions, and extension of the standard greedy set cover algorithm is beyond the scope of this article and is left to future work.

Second, we consider the problem of incomplete coverage. When all sensors can be used as permanent sensors, then the algorithm always terminates with complete coverage. This is because in the worst case every sensor can predict itself and so will eventually be added to P. However, when some sensors are excluded as permanent sensors, then it may not be possible to find a predictor for every sensor. In this case, the loop will terminate with U non-empty, and the user is given a warning (lines 10-11). There are several solutions for this problem. The unpredictable high-cost sensors can be included in the deployment anyway (reduce PX), or additional low-cost sensors can be deployed (increase S) or ϵ can be increased so they have more chance to be predicted by an existing physical sensor.

4.4 Fault-Tolerance and Stability

Algorithms 1 and 2 find a minimal set of physical sensors that covers the whole sensor field for a given error bound. However, a disadvantage is that the resulting network may not be fault-tolerant since virtual sensors may have only one predictor. Such sensors have no redundancy to allow for sensor failure. So if a physical sensor fails then the readings of all the virtual sensors it predicts are also lost. We now modify the selection algorithm to improve fault tolerance for a small cost increase of one physical sensor.

First, select P and V using Algorithm 2 and call these sets P_0 and V_0. Select $W \subseteq V_0$ as the set of virtual sensors that are predicted by only one permanent sensor.

$$W = \{w \in V_0 \mid size\ \{p \in P_0 \mid E[w,p] < \epsilon\} = 1\}.$$

Virtual sensors with only one predictor are the most vulnerable from a fault-tolerance viewpoint. Choose one virtual sensor $s_i \in W$ to become permanent. This should be the sensor whose predictions provide the best additional coverage of vulnerable virtual sensors.

$$s_i = max_{s_i \in W}\ \{size\ (\{s_j \mid E[i,j] < \epsilon\} \cap W)\ \}.$$

The final result is a new sensor partition with $P_1 = P_0 \cup \{s_i\}$ and $V_1 = V_0 \setminus \{s_i\}$. This process can be repeated to strengthen the fault tolerance of the virtual sensing system.

Stability is the ability of a system to maintain the accuracy of virtual sensors during long term operation. One reason for inaccurate predictions in the long term is because the context governing the measured phenomena has changed. For example, the performance of buildings is affected by seasonal changes in the weather. The other main reason for inaccurate predictions is ad hoc human actions. Human-caused changes could be passive effects of how the building is used (e.g. air flow and the number of residents) or unusual uses of artificial heating or cooling. They tend to be short term, and they are generally not predictable.

One advantage of our cross-correlation algorithm for learning virtual sensors is that when the context changes then both predictor and virtual sensor are likely affected and so the predictor will still be accurate. However, since context changes are, by definition, not always predictable in advance, BuildSense also needs to monitor the stability of virtual sensors in the long term.

BuildSense addresses this problem by monitoring the stability of its predictors during the operational phase. Selected permanent sensors, called **sentinels**, can be mirrored by other permanent sensors that can predict their values.

$$sentinels(W) = W \subset P\ \wedge\ \forall s_i \in W. \exists\, s_j \in P \setminus W.\ s_i \approx s_j.$$

Typically a permanent sensor set will yield several sentinels. Alternatively, additional permanent sensors can be added to increase the number of sentinels using a similar approach to the fault tolerance algorithm described above. Since both predicted and actual values are available for the sentinels, estimation errors can be monitored during operation. If the error for one or more sentinels drifts above a given bound for more than a few days, then the user can be warned that confidence in the virtual sensor predictions is low. The user may decide to trigger retraining with existing data to learn new predictors and offset functions. Additionally, but optionally, some temporary sensors may be redeployed and more data is gathered before the system is retrained. Alternatively, periods with high prediction error may be labeled as anomalous and this data simply treated with lower confidence. For example, one-off occupancy events such as heating peaks can be treated as anomalies that need not trigger retraining, while longer term changes caused by, for example, a change in the occupants of the house, can be addressed by triggering a retraining period, with or without gathering additional training data.

5 EVALUATION

The goal of BuildSense is to set up an easily manageable sensor infrastructure for any energy-efficient building. It advocates for a sensor network with minimal sensor deployment, yet supports a fine-grained data acquisition with high accuracy. To validate the framework, we address the following questions using the data collected from two real-world building monitoring networks.

Accuracy: How to ensure fine-grain data collection with the desired level of accuracy even if there is a sparse sensor deployment? Are the virtual sensor prediction functions of good quality?

Cost: What levels of cost saving are possible? Does the algorithm find optimal low-cost monitoring solutions?

Robustness: Is the network fault-tolerant to the failure of physical sensors? Are accurate predictions maintained for long operational periods?

Before evaluating BuildSense with respect to these performance criteria, let us first describe briefly the datasets that are used for the performance measurements.

5.1 Data Set

BuildSense is evaluated using data from two sensor networks that monitor energy-efficient rammed earth houses in the hot-arid climate of Kalgoorlie in Western Australia's goldfields [3]. The purpose of these networks is to investigate to what extent adopting passive solar design principles can reduce dependence on artificial climate control. The two single-floor houses have identical design and orientation, but one was built with the traditional solid rammed-earth walls (**monolithic**) and the other with an insulating polystyrene core into the walls (**insulated**).

The heterogeneous monitoring networks include commercial sensors for temperature and relative humidity, bespoke sensors including temperature sensor profilers embedded in 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. Figure 5 shows the floor plan and the placement of sensors in each of the houses, where KIT, LIV, BW, BS, BE refer to the Kitchen, Living Room, and Bedrooms West, South and East, respectively. The WASHING area comprises a toilet, bathroom, and laundry. The

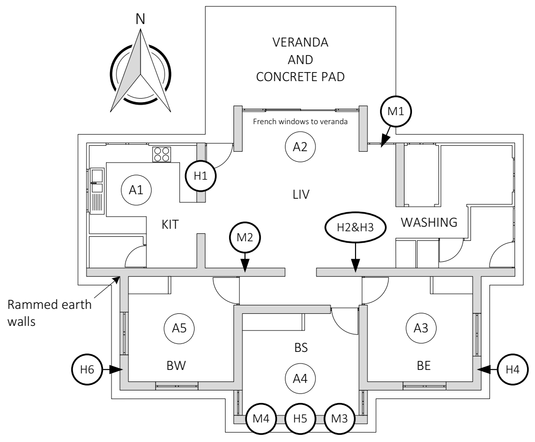

Figure 5: House plan and sensor placements for both the monolithic and insulated houses. See the text for an explanation of the sensor labels.

environs of the houses are instrumented with over 150 sensors. For each house, as shown in Figure 5, M1 to M4 each indicates profilers of 8 temperature sensors embedded within the rammed earth walls, and H1 to H6 indicate additional temperature sensors embedded in the walls. A1 to A5 are temperature and humidity sensors at ceiling height. In addition, A1 to A5 indicate the positions of 20 temporary head-level temperature and humidity sensors that were deployed before the building was occupied. Moreover, a weather station with 5 sensors is placed between the two houses. The monitoring sensor networks have been running continuously since September 2014 and the data has been made publicly available for future analyses [3].

For evaluation purposes, we partitioned the data into periods of one month for each house. Each month can be used either for training or testing. Both buildings were monitored but unoccupied from September to November 2014. The houses were each occupied by a family from December 2014 to June 2016. The number of active sensors varied slightly from month to month because – (a) a group of temporary head-level sensors was only available during the unoccupied period, and (b) communication errors or power failure caused some sensor streams to be lost for periods up to a few weeks. We used data only for those months where at least 60% of the sensors had sufficient data for training. The sensing intervals range from 5 minutes to 30 minutes. Linear interpolation was used to fill in up to 4 hours of missing values. Half hourly observations were then selected from each sensor stream, giving approximately 1440 observations per sensor per month.

5.2 Accuracy

BuildSense offers fine-grained sensing with minimal infrastructure with the help of virtual sensors, where virtual sensors are simply prediction models that replace physical sensors. Thus, the usability of BuildSense depends on the accuracy of the virtual sensor prediction models.

The prediction model used in BuildSense is a simple regression model in which the final virtual sensor readings are calculated by adding a per-hour offset to the predicted value. The hourly offsets used for virtual sensors are determined by a least squares approach that fits a model to the training data. Root mean squared error (RMSE) is used to specify the accuracy of a particular model as well as choosing the best model from available models. Despite being a simple model it is effective even for long term operation as demonstrated by the low overall RMSE values in Figure 6, and low per-month RMSEs in Figure 8. This section investigates the properties of BuildSense that contribute to its accuracy.

Consider a predictor sensor p for sensor s and a learned offset o so that the virtual sensor v is defined by $v = p + o^*$, where o^* maps the correct hourly offset at each point in the sequence. The sequence has length n. The residual r is a time series of the difference between the predicted and observed values for the virtual sensor: $r = v - s$.

The quality of the virtual sensor models is evaluated using the following standard metrics for the quality of regression models.

RMSE: $\sqrt{\Sigma(r \times r)/n}$. The RMSE should be as small as possible.

Homoscedasticity: The variance around the regression line should be the same for all values of the prediction variable. This can be measured by first partitioning the residuals into groups for each value of the prediction variable (e.g. to the nearest degree) and calculating the interquartile range (IQR) for each group. Homoscedasticity is measured by the standard deviation of these IQRs. Its value should be as close to 0 as possible.

Residual mean: $\Sigma\ r/n$. The mean of the residuals should be as close to 0 as possible.

Correlation: $cor(r, p)$. This measures the correlation between the physical sensor values and the residuals from the virtual sensor. Weak correlation indicates that the virtual sensor model is independent of the underlying context, whereas a strong correlation between these measures indicates that the model may be missing some parameter. We measure correlation using the pairwise Spearman correlation coefficient which ranges from -1 to 1. Values close to 0 indicate the weak correlation we desire.

Auto-correlation: represents recurring patterns in the residuals related to the time of observations. This can be calculated by grouping the residuals according to the hour of the day, calculating the interquartile range (IQR) and taking the standard deviation of the IQRs.

In order to investigate the quality of BuildSense prediction models, we investigate the performance of 17 virtual sensor pairs for the monolithic house with training data from the months of October and April combined. Virtual sensors are selected using an ϵ-neighborhood of 1.0° C: that is, all sensors that can be predicted with an error no more than 1 degree. We analyze readings from the complete data set of 22 months. Since there is missing data in this real world data set, the number of points with both estimated and actual data available ranges from 23,795 observations for the in-wall sensors to 3,708 observations for the five head-level sensors that could only be deployed before the houses were occupied. The remaining 12 sensor pairs have at least 16,616 points each.

Figure 6 shows the resulting quality metrics for each virtual sensor pair. The units are ° C for all metrics except the correlation coefficient, which is on the scale $[-1, 1]$. Sensor pairs were selected for RMSE $\leq$ 1.0° C in the training data. It can be seen that for all but one pair, the RMSE for the full data set of nearly 2 years is also

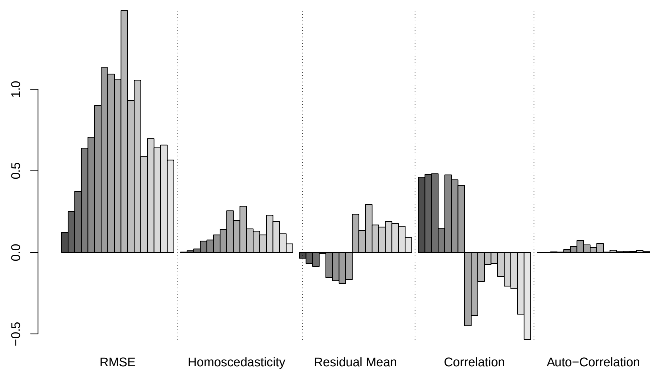

Figure 6: Virtual sensor residuals for 18 months.

within this limit. The variance around the regression line is close to 0 for all virtual sensor pairs as is the mean residual. Similarly, there is no significant bias of the residuals at different hours of the day. These metrics indicate that virtual sensor pairs remain accurate during long term operation. The correlation metric is somewhat surprising. Although the mean correlation coefficient is 0.0147 for all sensor pairs, there is high variance. The upper and lower quartiles are -0.223 and 0.445 respectively. That suggests that for many sensor pairs, as the temperature increases, the prediction error also rises. We considered whether this was because of the inclusion of some anomalous months. But running the same analysis with those months removed did not change the correlation results appreciably. We conclude that although there is some correlation between temperature and virtual sensor residuals, it does not adversely affect the overall accuracy of the system.

5.3 Cost

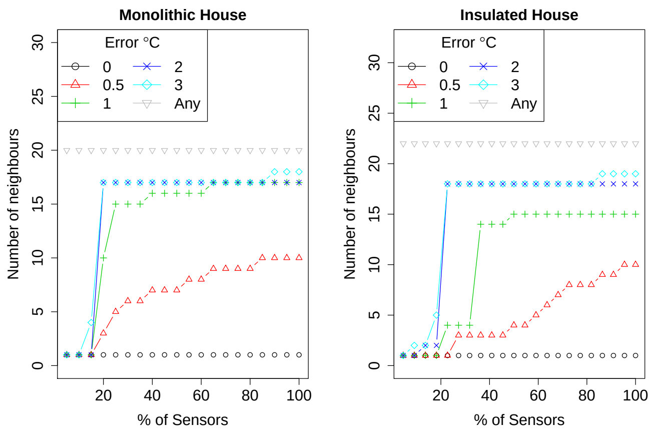

Figure 7: Number of ϵ-neighbors per sensor.

In this section, we evaluate how practical virtual sensing is in real-world scenarios by investigating the scope for replacing physical sensors with virtual ones. The size of sensors' ϵ-neighborhoods determines the scope for effective virtual sensing. Figure 7 shows the distribution of the size of sensor neighborhoods under different error bounds. As expected, higher error tolerance results in larger neighborhoods for more nodes, and so the scope for virtual sensing increases. Depending on the application, users can decide the appropriate balance between the number of physical sensors and the accuracy of the the virtual sensor predictions. For example, Figure 7 demonstrates that for an error bound of 1 degree (green line) 50% of the sensors have 16 sensors in their $\epsilon = 1$ neighborhood.

Table 2: Number of virtual sensors under differing selection criteria.

	Number of Virtual Sensors					
	All VS		P_0		P_1	
Error °C	Mono N=24	Ins N=25	Mono N=24	Ins N=25	Mono N=24	Ins N=25
0.5	54%	44%	33%	20%	38%	16%
1.0	71%	64%	0%	4%	54%	52%
2.0	71%	68%	0%	4%	67%	64%
3.0	75%	72%	25%	48%	67%	64%

Next, we evaluate how this works out in the selection of physical sensors. The choice of cost and accuracy parameters for sensor selection determines the number of virtual sensors that can be predicted. Table 2 gives the total costs of BuildSense networks under different selection criteria. Here we use a simple binary cost function of 1=permanent, 0=virtual. Column All VS shows the number of virtual sensors (as a percentage of all sensors) for two houses and error bounds from 0.5 to 3.0°C. Overall, for a realistic error requirement of 1.0° C, we achieved cost savings of 64% to 71% of deployed sensors in the two houses.

5.4 Robustness

Column P_0 of Table 2 shows the percentage of sensors that *are* fault-tolerant under the base selection algorithm. That is the number of sensors with more than one physical sensor as a predictor. The worst fault-tolerance occurs for the higher error bounds because they have larger neighborhoods and so a single physical sensor is typically able to predict most of the remaining sensors. To address this problem, we add an extra step, which selects one additional physical sensor in order to increase the predictor coverage of virtual sensors. Column P_1 of Table 2 shows that, for this additional cost of one extra physical sensor, we can significantly improve the proportion of fault-tolerant virtual sensors. For example, for error bound 1.0, the number of fault-tolerant sensors increases from 4% to 52%.

Next, we deal with the last question, i.e., what is the stability of the system in long run. In order to test stability, we choose October 2014 as training data and predict sensor values for the following 15 months of operation. The residuals between actual and predicted values in each month are recorded for each pair. Figure 8 shows the stability of predictions (RMSEs) per-month over the operational period for two virtual sensors and a sentinel selected from the training data. It can be seen that accuracy is not adversely affected by seasonal changes, which shows that the co-correlation model for virtual sensors works well. We conclude that, for this application,

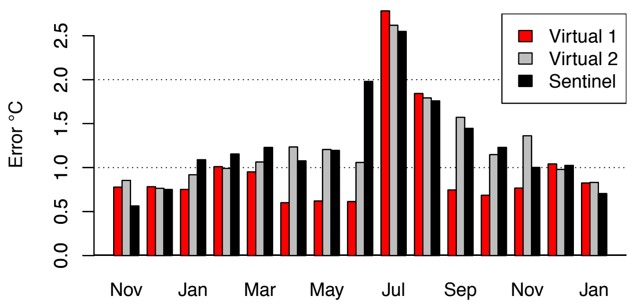

Figure 8: Stability of two virtual sensors and one sentinel as per-month RMSE over 14 months.

a training period of one month is sufficient for stable long term predictions with no need for retraining. However, the requirements for retraining depend strongly on the application. If the application context changes markedly or its accuracy requirements are very high, then users can opt for retraining. A third party can provide this service.

To summarize the results observed from the long term operation, the learned virtual sensor predictors are stable for more than one year of operations. There is one month (July) with errors ≥2° C and three months (June to August) ≥1.5° C, giving a stability ratio > 80%. In July and August (Australian Winter) there is a marked increase in prediction error. This is believed to be caused by the use of stand-alone electric heaters by the occupants, which is an ad hoc behavior that can not be predicted in advance. However, the sentinel clearly tracks this anomalous behavior. So sentinels can also serve as anomaly alarms.

6 CONCLUSION

Continuous, long-term, fine-grained, monitoring of buildings is essential for many approaches to energy-efficiency. Although deploying a large number of sensors throughout the building can provide an easy solution, it incurs a huge deployment and management cost. In this work, we propose BuildSense, a framework that enables continuous and fine-grained monitoring of a building with minimal sensor infrastructure. In many (typical) sensor networks, optimal sensor deployment may not be feasible due to inaccessibility of the monitoring field or lack of statistical knowledge of it. On the other hand, buildings offer an accessible and well-planned environment. However, they differ significantly from each other. Thus, planning an optimal and customized (at building-level) sensor deployment becomes a non-trivial task. Though we have used the datasets from two building monitoring networks of a special kind, the data itself does not contain any special characteristics. In other words, the dataset is sufficiently generic to maintain a generic evaluation of our framework. This ensures that BuildSense can be used for any building. Customization by building-specific physical sensor selection can be applied without any customization in the algorithm.

BuildSense uses a sense-learn-predict approach for customized sensor deployments. A large number of temporary or semi-permanent sensors are deployed during the learning phase. For the operational phase, the majority of these sensors are removed and a minimal number of sensors are selected for permanent monitoring. Using a mix of physical and virtual sensors, BuildSense ensures granular and continuous sensor data with sufficiently high accuracy and cost within a given budget. Overall, for a realistic error requirement of 1.0° C, we achieved cost savings of 64% and 71% of the deployed sensors in the two houses. On the other hand, for the more stringent accuracy requirements of 0.5 degree RMSE, a reduction of 44% and 54% of physical sensors can be achieved. We have also demonstrated that accurate and stable predictions can be maintained for over one year of operations.

There are several interesting avenues for future work. In this article, we used a simple binary cost function and a standard greedy set cover algorithm, but it would be interesting to experiment with different types of cost functions and study their effect on the optimal selection of virtual sensors. This scenario would inform improvements of the sensor selection algorithm. Another area for investigation would be incorporating convenience and trustworthiness into the selection criteria for virtual sensors. Small and unobtrusive sensors tend to be cheaper but also less accurate and reliable than more expensive sensors. Maybe using a set of small sensors could mask failures and so provide sufficiently accurate prediction at a lower cost than some expensive sensors. BuildSense has proved effective for temperature and humidity but further work is needed to assess its applicability for other sensed phenomena.

Acknowledgments

The authors acknowledge the support of the Australian Research Council (ARC Linkage Grant LP140100375), the Western Australian Department of Housing and Go8-DAAD. We thank the anonymous reviewers and paper shepherd for their valuable suggestions to improve the clarity of the paper. The research was approved by the Human Research Ethics Office of the University of Western Australia (RA/4/1/7273).

REFERENCES

[1] Bharathan Balaji, Hidetoshi Teraoka, Rajesh Gupta, and Yuvraj Agarwal. 2013. Zonepac: Zonal power estimation and control via hvac metering and occupant feedback. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings.* ACM, 1–8.

[2] Bharathan Balaji, Jian Xu, Anthony Nwokafor, Rajesh Gupta, and Yuvraj Agarwal. 2013. Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems.* ACM, 17.

[3] CTS Beckett, R Cardell-Oliver, D Ciancio, and C Huebner. 2017. Measured and simulated thermal behaviour in rammed earth houses in a hot-arid climate. Part B: Comfort. *Journal of Building Engineering* 13 (2017), 146–158.

[4] Mihaela Cardei, My T Thai, Yingshu Li, and Weili Wu. 2005. Energy-efficient target coverage in wireless sensor networks. In *INFOCOM 2005. 24th annual joint conference of the ieee computer and communications societies. proceedings ieee*, Vol. 3. IEEE, 1976–1984.

[5] Rachel Cardell-Oliver and Chayan Sarkar. 2016. Robust Sensor Data Collection over a Long Period Using Virtual Sensing. In *Proceedings of the Workshop on Time Series Analytics and Applications (TSAA '16).* 2–7.

[6] D. Chen. 2016. AccuRate and the Chenath engine for residential house energy rating. (2016). https://hstar.com.au/Home/Chenath Accessed: 2017-06-07.

[7] V. Chvatal. 1979. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research* 4, 3 (1979), 233–235. http://www.jstor.org/stable/3689577

[8] Declan T Delaney, Gregory MP O'Hare, and Antonio G Ruzzelli. 2009. Evaluation of energy-efficiency in lighting systems using sensor networks. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings.* ACM, 61–66.

[9] Amol Deshpande, Carlos Guestrin, Samuel R Madden, Joseph M Hellerstein, and Wei Hong. 2004. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30.* VLDB Endowment, 588–599.

[10] Varick L Erickson, Miguel Á Carreira-Perpiñán, and Alberto E Cerpa. 2011. OBSERVE: Occupancy-based system for efficient reduction of HVAC energy. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on.* IEEE, 258–269.

[11] Anca D Galasiu and Jennifer A Veitch. 2006. Occupant preferences and satisfaction with the luminous environment and control systems in daylit offices: a literature review. *Energy and Buildings* 38, 7 (2006), 728–742.

[12] Ali Ghahramani, Farrokh Jazizadeh, and Burcin Becerik-Gerber. 2014. A knowledge based approach for selecting energy-aware and comfort-driven HVAC temperature set points. *Energy and Buildings* 85 (2014), 536–548.

[13] Carlos Guestrin, Peter Bodik, Romain Thibaux, Mark Paskin, and Samuel Madden. 2004. Distributed regression: an efficient framework for modeling sensor network data. In *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on.* IEEE, 1–10.

[14] Himanshu Gupta, Vishnu Navda, Samir Das, and Vishal Chowdhary. 2008. Efficient gathering of correlated data in sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 4, 1 (2008), 4.

[15] Hongbo Jiang, Shudong Jin, and Chonggang Wang. 2011. 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 (2011), 1064–1071.

[16] Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. 2006. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th international conference on Information processing in sensor networks.* ACM, 2–10.

[17] Andrew Krioukov, Stephen Dawson-Haggerty, Linda Lee, Omar Rehmane, and David Culler. 2011. A living laboratory study in personalized automated lighting controls. In *Proceedings of the third ACM workshop on embedded sensing systems for energy-efficiency in buildings.* ACM, 1–6.

[18] Chong Luo, Feng Wu, Jun Sun, and Chang Wen Chen. 2009. Compressive data gathering for large-scale wireless sensor networks. In *Proceedings of the 15th annual international conference on Mobile computing and networking.* ACM, 145–156.

[19] S Mini, Siba K Udgata, and Samrat L Sabat. 2014. Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sensors Journal* 14, 3 (2014), 636–644.

[20] U.S. Department of Energy (DOE). 2008. Buildings Energy Data Book. http://www.c2es.org/technology/overview/buildings. (2008). Accessed: 2017-05-22.

[21] Devika Pisharoty, Rayoung Yang, Mark W Newman, and Kamin Whitehouse. 2015. Thermocoach: Reducing home energy consumption with personalized thermostat recommendations. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments.* ACM, 201–210.

[22] Maher Rebai, Hichem Snoussi, Faicel Hnaien, Lyes Khoukhi, and others. 2015. Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. *Computers & Operations Research* 59 (2015), 11–21.

[23] Silvia Santini and Kay Romer. 2006. 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).* 29–36.

[24] Chayan Sarkar, Vijay S Rao, and R Venkatesha Prasad. 2014. No-sense: Sense with dormant sensors. In *Communications (NCC), 2014 Twentieth National Conference on.* IEEE, 1–6.

[25] Chayan Sarkar, Vijay S Rao, R Venkatesha Prasad, Sankar Narayan Das, Sudip Misra, and Athanasios Vasilakos. 2016. VSF: An Energy-Efficient Sensing Framework Using Virtual Sensors. *IEEE Sensors Journal* 16, 12 (2016), 5046–5059.

[26] Chayan Sarkar, Akshay Uttama Nambi SN, and R Venkatesha Prasad. 2016. iLTC: Achieving Individual Comfort in Shared Spaces. In *Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN).* ACM.

[27] Mina Sartipi and Robert Fletcher. 2011. Energy-efficient data acquisition in wireless sensor networks using compressed sensing. In *Data Compression Conference (DCC), 2011.* IEEE, 223–232.

[28] Di Tian and Nicolas D Georganas. 2003. A node scheduling scheme for energy conservation in large wireless sensor networks. *Wireless Communications and Mobile Computing* 3, 2 (2003), 271–290.

[29] Liu Xiang, Jun Luo, and Athanasios Vasilakos. 2011. Compressed data aggregation for energy efficient wireless sensor networks. In *Sensor, mesh and ad hoc communications and networks (SECON), 2011 8th annual IEEE communications society conference on.* IEEE, 46–54.