8-28-2012

Autonomous Correction of Sensor Data Applied to Building Technologies Utilizing Statistical Processing Methods

Charles C. Castello
Oak Ridge National Laboratory, castellocc@ornl.gov

Joshua New
Oak Ridge National Laboratory, newjr@ornl.gov

芽|Sprouts Working Papers on Information Systems ISSN 1535-6078

Autonomous Correction of Sensor Data Applied to Building Technologies Utilizing Statistical Processing Methods

Charles C. Castello
Oak Ridge National Laboratory, USA

Joshua New
Oak Ridge National Laboratory, USA

Abstract
Autonomous detection and correction of potentially missing or corrupt sensor data is a essential concern in building technologies since data availability and correctness is necessary to develop accurate software models for instrumented experiments. Therefore, this paper aims to address this problem by using statistical processing methods including: (1) least squares; (2) maximum likelihood estimation; (3) segmentation averaging; and (4) threshold based techniques. Application of these validation schemes are applied to a subset of data collected from Oak Ridge National Laboratory's (ORNL) ZEBRAlliance research project, which is comprised of four single-family homes in Oak Ridge, TN outfitted with a total of 1,218 sensors. The focus of this paper is on three different types of sensor data: (1) temperature; (2) humidity; and (3) energy consumption. Simulations illustrate the threshold based statistical processing method performed best in predicting temperature, humidity, and energy data.

Keywords: Sensor data validation; statistical processing methods; least squares; maximum likelihood estimation; segmentation averaging; threshold based; building technologies.

Permanent URL: http://sprouts.aisnet.org/12-6

Reference: Castello, C. , New, J. (2012). "Autonomous Correction of Sensor Data Applied to Building Technologies Utilizing Statistical Processing Methods," Proceedings > Proceedings of Energy Informatics . *Sprouts: Working Papers on Information Systems*, 12(6). http://sprouts.aisnet.org/12-6

LIST OF NOMENCLATURE

Abbreviations

GHG	Greenhouse gas
K-SOM	Kohonen self-organizing map
MLE	Maximum likelihood estimation
ORNL	Oak Ridge National Laboratory
PCA	Principal component analysis
RMSE	Root-mean-square error
USGBC	U.S. Green Building Council
WAHP	Water-to-air heat pumps
WWHP	Water-to-water heat pumps

Symbols

μ	Mean
σ	Standard deviation
$\hat{\sigma}$	Maximum likelihood estimate
C	Standard deviation multipliers
c	Element of C
d	Degree of polynomial
e_{rel}	Relative error
e_{abs}	Absolute error
f	Predicted value
g	Gaussian distribution
$histMean$	Historical mean
$histStd$	Historical standard deviation
L	Least squares estimation
m	Expected value
$\widehat{m_{SA}}$	Segmentation average
N	Number of samples
n	current time-step
o	Size of observation window
P	Coefficients of polynomial
p	Element of P
r	Residual
X	Independent variables
x	Element of X
s	First time-step of observation window
Y	Sensor data
y	Element of Y

INTRODUCTION

Energy consumption in the U.S. is a critical area of concern where residential and commercial buildings consume approximately 40% of total primary energy (U.S. Department of Energy, 2008). Retrofitting inefficient existing buildings with new and innovative technologies

that help to curb energy consumption will ensure the reduction of energy consumption and enhance the ability to optimize use of our energy distribution infrastructure (Miller et al., 2011). Buildings also have the best potential for reducing green-house-gas (GHG) emissions since the building sector exceeds both the industrial and transportation sectors in the U.S. (Intergovernmental Panel on Climate Change, 2007). There is a need for integrated building strategies, according to the U.S. Green Building Council (USGBC), in order to achieve Net Zero Energy buildings (U.S. Green Building Council Research Committee, 2007). Therefore, the conservation of energy and mitigation of GHG emissions hinges on the continued research of energy efficient buildings and technologies.

There is much research dealing with the improvement of energy efficiency in commercial buildings and residential homes (Christian, 2010; Norton and Christensen, 2006; Miller and Kosny, 2008; Parker et al., 2011). These include several fundamental concerns relevant to sensors being used to collect a wide variety of variables (e.g., humidity ratio, solar flux, temperature, time, wind speed, etc.) in order to analyze and understand the capabilities of components, systems, and whole-buildings for enhanced energy efficiency. Based on the number of variables being collected and sampling rates, the amount of data being assembled has the potential of being large-scale. An example of this is the ZEBRAlliance research project, which in 2008 built four residential homes to be used for integration of Oak Ridge National Laboratory's (ORNL's) energy-efficient technologies to gauge the integral success and affordability of components and houses (ZEBRAlliance, 2008). The first and second homes consist of 279 sensors, the third home has 321 sensors, and the fourth home has 339 sensors, a majority of which are measuring temperature (thermistors, thermocouples, and combo probes), relative humidity (RH and combo probes), and electrical usage (watt-meters). Most sensors have a 15-minute resolution with approximately 80 sensors having a resolution of 1-minute, although hourly, daily, and monthly reports are also consolidated. There are 9,352 data points in an hour, 224,448 in a day, 1,571,136 in a week, and 81,699,072 in a year. Many concerns arise with this amount of data points being collected in such a real-world experiment, specifically data corruption from sensor failure, sensor fouling, calibration error, and data logger failure.

Sensor validation is vital for energy efficiency research and control in buildings. Even with the most sophisticated instruments and control systems, analysis and decisions based on faulty data could lead to inaccuracies when dealing with components, systems, and whole-buildings for improved energy efficiency. There are currently two approaches that are widely used for the validation of data: analytical redundancy and hardware redundancy (Ibarguengoytia et al., 2001). Analytical redundancy uses mathematical relationships between measurements to predict a sensor's value. When the number of sensors and the complexity of the model increase, the analytical redundancy approach becomes inefficient. Another disadvantage of the analytical redundancy approach is that each derived relationship is very specific to the data; meaning a slight modification may require significant resources to stabilize. Hardware redundancy on the other hand is not always possible due to the need for increased sensors, data acquisition channels/systems, installation and maintenance labor, etc.). Therefore, this research aims to use independent relationships (e.g., interpolation based on available data from a single sensor), instead of dependent relationships (e.g., prediction using data from other sensors), and statistical processing methods. Sensor calibrations and manufacturer's rated accuracy is not considered in this work. Calculating the predicted value of a sensor as a function of others and assuming or leveraging periodic patterns in the data is also not covered in this research.

The statistical processing methods that are investigated in this paper are: (1) least squares; (2) maximum likelihood estimation; (3) segmentation averaging; and (4) threshold based techniques for sensor data validation. These procedures are used for data prediction which is compared with original data to determine the accuracy of each method and is applied to ZEBRAlliance sensor data for temperature, humidity, and energy consumption. Results from this study show the data to be best predicted using the threshold based statistical processing method.

BACKGROUND

There has been a rich variety of research dealing with data validation for numerous applications (Ibarguengoytia et al., 2001; Frolik et al, 2001; Uluyol et al., 2006; Postolache, 2005). In Ibarguengoytia et al., 2001, two Bayesian networks were used for the detection of faults in a set of sensors; the first represents the dependencies among all sensors and the second isolates the faulty sensor. Self-validation, fusion, and reconstruction of sensor data was tackled in Frolik et al, 2001 by exploring three key steps: (1) employ fuzzy logic rules for self-validation and self-confidence; (2) exploit near-linear relationships between sensors for reconstructing missing or low-confidence data; and (3) fuse this data into a single measurement along with a qualitative indicator for its reliability. A start-up fault detection and diagnosis method was presented for gas turbine engines in (Uluyol et al., 2006), which consisted of three key techniques: (1) statistics; (2) signal processing; and (3) soft computing. Sensor profiles were generated from good and bad engine start-ups in which a feature vector was calculated and signal processing was used for feature selection. In the signal-processing step, principal component analysis (PCA) was applied to reduce the samples consisting of sensor profiles into a smaller set. The features obtained from this step were then classified using neural-network-based methods. In Postolache, 2005, a Kohonen self-organizing map (K-SOM) was used to perform sensor data validation and reconstruction. Sensor failure and pollution event detections were also studied with the use of this methodology for a water quality sensor network application.

There has been a wide range of work in regards to sensor data validation using not only statistical methods, but also filtering and machine learning techniques as well. However, all previously mentioned research deals with dependent relations among multiple sensors for validation. Dependencies in data prediction require greater computational resources and datasets. Independent data validation conserves these resources by requiring past data for a given sensor, lending itself to greater parallel throughput and scalability. Therefore, this paper uses statistical processing methods for independent data validation applied to building technologies.

STATISTICAL PROCESSING METHODS

The following statistical methods from Bo et al., 2009 detailed the use of statistical techniques to predict wireless field strength. The four methods discussed: (1) least squares; (2) maximum likelihood estimation; (3) segmentation averaging; and (4) threshold based, are modified to meet the needs of fault detection and sensor data prediction. Artificial gaps are introduced by randomly removing portions of existing data for testing the accuracy of auto-correcting algorithms. This is accomplished by randomly generating training and testing subsets. In this paper, training and testing subsets are split 70% and 30% respectively. Each sensor is used as an independent variable and predicts sensor values based upon a variable-sized window of observations. A prediction model is generated for each window of observations and auto-

correction (interpolation and/or extrapolation) occurs if values are missing or corrupt (far away from the predicted value), though original data is always preserved for reference. This paper does not take into consideration when all elements in a given window are missing and/or corrupt. This scenario can be handled using previous and subsequent models generated from windows with enough elements for predictions (i.e., using multiple windows to generate a model instead of elements within a window).

For all statistical processing techniques in this research, an observation window of size *o* is used to predict the sensor's data value for each time-step within the observation window. The observation window moves forward by *o* time-steps (no overlap) and prediction for each sample within the observation window is calculated. This process occurs for every possible window within a given set of time-series sensor data. Root-mean-square error (RMSE), relative error, and absolute error, Equations 11, 12, and 13 respectively, are calculated for each prediction to determine the performance.

To give the reader a better understanding of observation windows, an example of its use is given. Let's say least squares method is used for validation purposes (reviewed in the following section) with the observation window being of size *o*=24. Temperature data is used which is randomly split into training (70%) and testing (30%) subsets. During training, each observation window generates a model based on the training samples in that window. The model is used to predict the behavior of temperature in that observation window. An example in Figure 1 shows the 10th observation window in the temperature dataset (left). This model is then used to predict at time-steps where testing data is located (right).

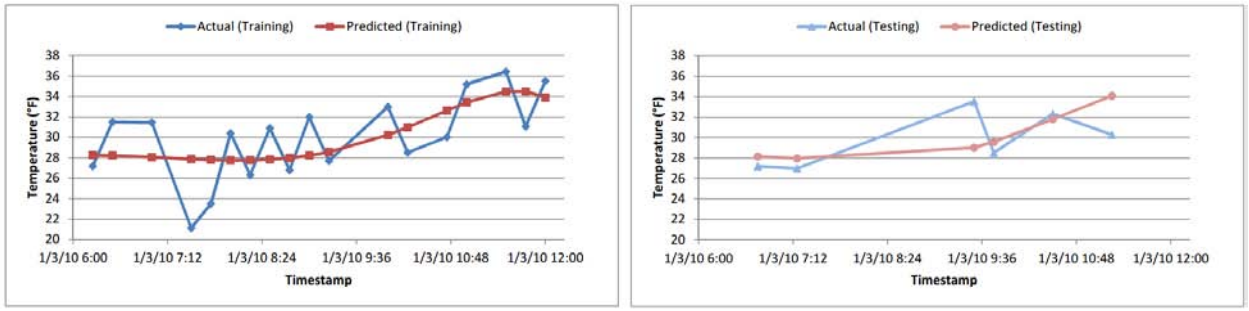

Figure 1. Actual and predicted temperature values of training (left) and testing (right) subset.

Least Squares

Calculating least square estimation is accomplished through calculating the squared residual between the inputs and the predicted values and summing. This is shown as:

$$L = \sum_{s=n-o}^{n-1} [y(s) - f(s)]^2 \quad (1)$$

where *n* is the current time-step, *s* represents the time-step relative to the observation window, *y*(*s*) is actual sensor data, and *f*(*s*) signifies the predicted value. Polynomial fitting is used to predict data values based on a learned model of data in the observation window. Polynomial fitting is achieved by finding (*d*+1) coefficients, *P*, of a *d*th degree polynomial. A generalized form of the polynomial is:

$$f = p_1 + p_2 x + \cdots + p_{d+1} x^d \quad (2)$$

where x is the independent variable (i.e., time-step) and f is the dependent variable (i.e., prediction). The polynomial in its generalized matrix form is shown as:

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^d \\ 1 & x_2 & \cdots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & \cdots & x_{n-1}^d \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \tag{3}$$

The matrix notation for polynomial fit is:

$$f = \mathbf{X}P \tag{4}$$

The coefficients of the polynomial, P, can be solved by multiplying both sides of Equation (4) by the transpose X^T:

$$\mathbf{X}^T f = \mathbf{X}^T \mathbf{X} P \tag{5}$$

Therefore, the coefficients of the polynomial, P, are:

$$P = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T f \tag{6}$$

Maximum Likelihood Estimation

The maximum likelihood estimation is calculated using the Gaussian distribution (Harris and Stocker, 1998; Hoel, 1984), which is assumed in this research for temperature, humidity, and energy data. For a Gaussian distribution:

$$g(y_1,...,y_s \mid m,\sigma) = \prod \frac{1}{\sigma\sqrt{2\pi}} e^{-(y_s-m)^2/(2\sigma^2)} = \frac{(2\pi)^{-\pi/2}}{\sigma^s} \exp\left[-\frac{\sum (y_s - m)^2}{2\sigma^2} \right] \tag{7}$$

where m is the expected value, σ is the standard deviation, $y(s)$ represents actual sensor data, and s represents the time-step. Therefore, the maximum likelihood estimate is:

$$\hat{\sigma} = E(y) = \sqrt{\frac{\sum (y_s - \mathrm{m})^2}{\mathrm{s}}} \tag{8}$$

Segmentation Averaging

In segmentation averaging, a window is set for the smoothing average. The window length (i.e., number of observations) is denoted as o and the sensor data is represented by y in the segmentation averaging process which can be express as:

$$\widehat{m_{SA}} = \frac{\left[\frac{(y_1 + \cdots + y_{n-1})}{o} + \frac{(y_2 + \cdots + y_{n-1})}{o} + \cdots + \frac{(y_{N-o+1} + \cdots + y_N)}{o} \right]}{N - o + 1} \tag{9}$$

where n is the current time-step and N is the total number of samples.

Threshold Based

The threshold based method uses a threshold value to determine whether or not the sensor data is too large or too small for averaging. In this research work, the threshold is defined as:

$$threshold = \mu + c\sigma \tag{10}$$

where μ is the mean, σ is the standard deviation, and c is the standard deviation multiplier which controls the threshold value. Any sensor data value, y, within the threshold, is used to calculate a moving average, *predMean*, based on the o observations. The *predMean* value is then used as the prediction for all time-steps in the observation window.

EXPERIMENTAL DATASET

The experimental dataset for this research is taken from ORNL's ZEBRAlliance project (ZEBRAlliance, 2008), specifically temperature, humidity, and energy usage sensor data from house #2 of four during the 2010 calendar year. House #2 consists of high efficiency technologies, specifically advanced framing for its envelopes, high-efficiency florescent lighting, and Energy Star appliances. Space conditioning is provided by water-to-air heat pumps (WAHP). Water heating is provided by special build water-to-water heat pumps (WWHP). The temperature and humidity data is taken from the energy recovery ventilation (ERV) unit's outside intake ("Z09_T_ERV_IN_Avg" and "Z09_RH_ERVin_Avg" respectively). Units for temperature and humidity are degrees Fahrenheit (°F) and percentage of relative humidity (%RH) respectively. The energy usage data is taken from the home's refrigerator ("A01_WH_fridge_Tot"). The unit for energy usage is Watt-hour (Wh). Data was collected through Campbell Scientific's CR1000 measurement and control datalogger. The resolution of all three data types is 15 min giving a total number of samples for each sensor, N = 35,040.

EXPERIMENTAL SETUP

Simulations using statistical processing methods include least squares, maximum likelihood estimation, segmentation averaging, and threshold based applied to temperature, humidity, and energy data. Procedures for each technique are discussed to understand how results are generated. Pseudo-code for each method is also given.

The performance metrics used for statistical processing methods are root-mean-square error (RMSE), relative error, and absolute error. RMSE is calculated by:

$$RMSE = \sqrt{\frac{1}{o}(r_1^2 + r_2^2 + \cdots + r_o^2)} \tag{11}$$

where r_s^2 represents a residual difference between the actual sensor value and the predicted value. The relative error is calculated by:

$$e_{rel,o,n} = \sum_{s=n-o}^{n-1} \left|\frac{r(s)}{y(s)}\right| \tag{12}$$

where n is the current time-step, s represents the first time-step of the observation window, $y(s)$ is actual sensor data, and $r(s)$ is the residual corresponding to $y(s)$. The absolute error is calculated by:

$$e_{abs,o,n} = \sum_{s=n-o}^{n-1} \left| \frac{r(s)}{y_{max} - y_{min}} \right| \quad (13)$$

where y_{max} and y_{min} are the maximum and minimum sensor data values respectively within the sensor dataset, Y.

Least Squares

The pseudo-code for least squares statistical processing method is shown in Figure 2. The inputs of this technique are the sensor dataset, Y and the observation window's size, o, which are

1 **Input:** # of observations taken into account (o) and input dataset (Y)
2 **Output:** Root-mean-square error (RMSE), ε_{mean}, relative error, $E_{rel\text{-}mean}$, and absolute error, $E_{abs\text{-}mean}$
3 **begin**
4 Randomly divide dataset Y into training set Y_{train} (70%) and test set Y_{test} (30%)
5 *// training*
6 $m = 1$ *// variable to keep track of starting point of observations used in prediction*
7 *// loop through all input values where* $Y_{train} = \{y_1, y_2, \ldots, y_{(0.70)*N}\}$
8 **for** $j = o$ to (0.30)*N **do** *// N=size(Y) and iteration of o*
9 $deg = o$
10 Calculate coefficients, $P = \{p_1, p_2, \ldots, p_{k+1}\}$, of polynomial fit for data $Y(m:j)$, degree deg using Equation (6)
11 Record coefficients P in P_{coll}
12 Calculate predicted values using P with Equation (2)
13 Calculate residuals, $R = \{r_1, r_2, \ldots, r_o\}$
14 Calculate RMSE, ε using Equation (11)
15 Record ε value in $\varepsilon_{coll} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{(0.70)*N-o}\}$
16 Calculate relative error, e_{rel} using Equation (12)
17 Record e_{rel} value in $E_{rel\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.70)*N-o}\}$
18 Calculate absolute error, e_{abs} using Equation (13)
19 Record e_{abs} value in $E_{abs\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.70)*N-o}\}$
20 $m = m + o$ *// iterate*
21 **end**
22 Calculate the mean value of ε_{coll}, ε_{mean}
23 Calculate the mean value of $E_{rel\text{-}coll}$, $E_{rel\text{-}mean}$
24 Calculate the mean value of $E_{abs\text{-}coll}$, $E_{abs\text{-}mean}$
25 ε_{mean}, $E_{rel\text{-}mean}$, and $E_{abs\text{-}mean}$ = null
26 *// testing*
27 $m = 1$ *// variable to keep track of starting point of observations used in prediction*
28 *// loop through all input values where* $Y_{test} = \{y_1, y_2, \ldots, y_{(0.30)*N}\}$
29 **for** $j = o$ to (0.30)*N **do** *// N=size(Y) and iteration of o*
30 Calculate predicted values using P coefficients retrieved from P_{coll}
31 Calculate residuals, $R = \{r_1, r_2, \ldots, r_o\}$
32 Calculate RMSE, ε using Equation (11)
33 Record ε value in $\varepsilon_{coll} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{(0.30)*N-o}\}$
34 Calculate relative error, e_{rel} using Equation (12)
35 Record e_{rel} value in $E_{rel\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.30)*N-o}\}$
36 Calculate absolute error, e_{abs} using Equation (13)
37 Record e_{abs} value in $E_{abs\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.30)*N-o}\}$
38 $m = m + o$ *// iterate*
39 **end**
40 Calculate the mean value of ε_{coll}, ε_{mean}
41 Calculate the mean value of $E_{rel\text{-}coll}$, $E_{rel\text{-}mean}$
42 Calculate the mean value of $E_{abs\text{-}coll}$, $E_{abs\text{-}mean}$
43 ε_{mean}, $E_{rel\text{-}mean}$, and $E_{abs\text{-}mean}$ = null

Figure 2. Algorithm of least squares experimental setup.

used to determine a model for the data within the window. The outputs are the calculated performance metrics, specifically RMSE, relative error, and absolute error. The sensor dataset, Y, is randomly split into a training set (70%) and test set (30%). The training set is used to determine the model for each observation window. These models are then used with the test set to determine the prediction accuracy of least squares.

The data samples in the observation window are used to calculate coefficients in Equation (6) for polynomial curve fitting. Degree, $d = o$, is used to calculate the coefficients, P, for polynomial curve fitting. The calculated curve based on P is used to calculate the predicted o values within the observation window using Equation (2). The residuals are then calculated for each observation window within the sensor dataset, Y. The results section and accompanying tables show the performance metrics for all observation windows within the dataset, Y, for the sensor data.

Maximum Likelihood Estimation

The pseudo-code for maximum likelihood estimation is shown in Figure 3. The inputs of this

```
Input: # of observations taken into account (o) and input dataset (Y)
Output: Root-mean-square error (RMSE), ε_mean, relative error, E_rel-mean, and absolute error, E_abs-mean
begin
Randomly divide dataset Y into training set Y_train (70%) and test set Y_test (30%)
// training
m = 1 // variable to keep track of starting point of observations used in prediction
// loop through all input values where Y_train = {y_1, y_2, ..., y_(0.70)*N}
for j = o to (0.30)*N do // N=size(Y) and iteration of o
    Calculate the maximum likelihood estimate using the input Y(m:j) using Equation (8)
    Record maximum likelihood estimate into m_coll
    Calculate residuals, R = {r_1, r_2, ..., r_o}
    Calculate RMSE, ε using Equation (11)
    Record ε value in ε_coll = {ε_1, ε_2, ..., ε_(0.70)*N-o}
    Calculate relative error, e_rel using Equation (12)
    Record e_rel value in E_rel-coll = {e_1, e_2, ..., e_(0.70)*N-o}
    Calculate absolute error, e_abs using Equation (13)
    Record e_abs value in E_abs-coll = {e_1, e_2, ..., e_(0.70)*N-o}
    m = m + o // iterate
end
Calculate the mean value of ε_coll, ε_mean
Calculate the mean value of E_rel-coll, E_rel-mean
Calculate the mean value of E_abs-coll, E_abs-mean
ε_mean, E_rel-mean, and E_abs-mean = null
// testing
m = 1 // variable to keep track of starting point of observations used in prediction
// loop through all input values where Y_test = {y_1, y_2, ..., y_(0.30)*N}
for j = o to (0.30)*N do // N=size(Y) and iteration of o
    Calculate predicted values using maximum likelihood estimated retrieved from m_coll
    Calculate residuals, R = {r_1, r_2, ..., r_o}
    Calculate RMSE, ε using Equation (11)
    Record ε value in ε_coll = {ε_1, ε_2, ..., ε_(0.30)*N-o}
    Calculate relative error, e_rel using Equation (12)
    Record e_rel value in E_rel-coll = {e_1, e_2, ..., e_(0.30)*N-o}
    Calculate absolute error, e_abs using Equation (13)
    Record e_abs value in E_abs-coll = {e_1, e_2, ..., e_(0.30)*N-o}
    m = m + o // iterate
end
Calculate the mean value of ε_coll, ε_mean
Calculate the mean value of E_rel-coll, E_rel-mean
Calculate the mean value of E_abs-coll, E_abs-mean
ε_mean, E_rel-mean, and E_abs-mean = null
```

Figure 3. Algorithm of MLE experimental setup.

technique are the sensor dataset, Y, and the window size, o. The sensor dataset, Y, is randomly split into a training set (70%) and test set (30%). The training set is used to determine the model for each observation window. These models are then used with the test set to determine the prediction accuracy of maximum likelihood estimation. The number of observations being studied, o, in the observation window is used to calculate the maximum likelihood estimate shown in Equation (8). The maximum likelihood estimate is calculated for all observation windows of size o for the sensor dataset, Y. The calculated maximum likelihood estimate is the predicted value for all o observations within the window which are used to calculate the residuals.

Segmentation Averaging

The pseudo-code for segmentation averaging is shown in Figure 4. The inputs of this

Input: # of observations taken into account (o) and input dataset (Y)
Output: Root-mean-square error (RMSE), ε_{mean}, relative error, $E_{rel\text{-}mean}$, and absolute error, $E_{abs\text{-}mean}$
begin
Randomly divide dataset Y into training set Y_{train} (70%) and test set Y_{test} (30%)
// training
$m = 1$ *// variable to keep track of starting point of observations used in prediction*
// loop through all input values where $Y_{train} = \{y_1, y_2, \ldots, y_{(0.70)*N}\}$
for $j = o$ to (0.30)*N **do** *// N=size(Y) and iteration of o*
  Calculate segmentation average, $avgSeg = Y(m:j)$ using Equation (9)
  Record $avgSeg$ value in $avgSeg_{Coll} = \{avgSeg_1, avgSeg_2, \ldots, avgSeg_{N\text{-}initialObs}\}$
  Calculate mean segmentation average, msa = mean($avgSeg_{Coll}$)
  Record msa into msa_{coll}
  Calculate residuals, $R = \{r_1, r_2, \ldots, r_o\}$
  Calculate RMSE, ε using Equation (11)
  Record ε value in $\varepsilon_{coll} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{(0.70)*N-o}\}$
  Calculate relative error, e_{rel} using Equation (12)
  Record e_{rel} value in $E_{rel\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.70)*N-o}\}$
  Calculate absolute error, e_{abs} using Equation (13)
  Record e_{abs} value in $E_{abs\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.70)*N-o}\}$
  $m = m + o$ *// iterate*
end
Calculate the mean value of ε_{coll}, ε_{mean}
Calculate the mean value of $E_{rel\text{-}coll}$, $E_{rel\text{-}mean}$
Calculate the mean value of $E_{abs\text{-}coll}$, $E_{abs\text{-}mean}$
ε_{mean}, $E_{rel\text{-}mean}$, and $E_{abs\text{-}mean}$ = null
// testing
$m = 1$ *// variable to keep track of starting point of observations used in prediction*
// loop through all input values where $Y_{test} = \{y_1, y_2, \ldots, y_{(0.30)*N}\}$
for $j = o$ to (0.30)*N **do** *// N=size(Y) and iteration of o*
  Calculate predicted values using msa retrieved from msa_{coll}
  Calculate residuals, $R = \{r_1, r_2, \ldots, r_o\}$
  Calculate RMSE, ε using Equation (11)
  Record ε value in $\varepsilon_{coll} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{(0.30)*N-o}\}$
  Calculate relative error, e_{rel} using Equation (12)
  Record e_{rel} value in $E_{rel\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.30)*N-o}\}$
  Calculate absolute error, e_{abs} using Equation (13)
  Record e_{abs} value in $E_{abs\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.30)*N-o}\}$
  $m = m + o$ *// iterate*
end
Calculate the mean value of ε_{coll}, ε_{mean}
Calculate the mean value of $E_{rel\text{-}coll}$, $E_{rel\text{-}mean}$
Calculate the mean value of $E_{abs\text{-}coll}$, $E_{abs\text{-}mean}$
ε_{mean}, $E_{rel\text{-}mean}$, and $E_{abs\text{-}mean}$ = null

Figure 4. Algorithm of segmentation averaging experimental setup.

technique are the sensor dataset, *Y*, and the number of observations, o. As before, the data is split into a training set (70%) and test set (30%). The number of observations being studied, *o*, in the observation window is used to calculate the segmentation average in Equation (9) for each sample in the observation window. The calculated segmentation average is used as the predicted value for all *o* observations within the window and used to calculate the residuals. The segmentation average is calculated for all observation windows of size *o* for the sensor dataset, *Y*. Results show the mean RMSE, relative error, and absolute error values for all of the observation windows within the dataset, *Y*, for temperature, humidity, and energy data.

Threshold Based

The pseudo-code for threshold based statistical processing method is shown in Figure 5. The standard deviation (*histStd*), and the number of standard deviations used to calculate the threshold (*c*). The outputs are the calculated performance metrics which are reported in the results section. The sensor dataset, *Y*, is randomly split into a training set (70%) and test set (30%). The training set is used to determine the model for each observation window. These models are then used with the test set to determine the prediction accuracy of threshold based averaging. The input values within the observation window of size *o* are compared with a specified threshold. The threshold is calculated using Equation (10). The threshold is calculated for all observation windows of size *o* for the sensor dataset, *Y*. This study takes into consideration $C = \{1, 2, 3\}$ which signifies investigating data within one, two, and three standard deviations (σ) away from the mean (μ). If the observation is less than or equal to the threshold value, the value is used in an average, which is used as the predicted value for all *o* observations within the window and to calculate the residuals.

Input: # of observations taken into account (o),input dataset (Y), historical mean of the input (*histMean*), historical standard deviation of the input (*histStd*), and number of standard deviations used to calculate the threshold (c)
Output: Root-mean-square error (RMSE), ε_{mean} , relative error, $E_{rel\text{-}mean}$, and absolute error, $E_{abs\text{-}mean}$
begin
Randomly divide dataset Y into training set Y_{train} (70%) and test set Y_{test} (30%)
Calculate the threshold, *threshold* = *histMean* + (*c* * *histStd*) using Equation (10)
// training
$m = 1$ *// variable to keep track of starting point of observations used in prediction*
// loop through all input values where $Y_{train} = \{y_1, y_2, \ldots, y_{(0.70)*N}\}$
for $j = o$ to (0.30)*N **do** *// N=size(Y) and iteration of o*
    // loop through all input values under consideration, $Y_{train}(m{:}j)$
    for k = m to j **do**
        // check if Y(k) is less than or equal to the calculated threshold value
        if $Y(k) \leq threshold$
            Record $Y(k)$ value in $Y_{thres} = \{y_1, y_2, y_3, \ldots, y_P\}$ *// P is the number of inputs that are less than or equal to the calculated threshold value*
        end
    end
Y_{thres} = null
Calculate the predicted value based on the threshold, *predMean* = mean(Y_{thres})
Record *predMean* into $predMean_{coll}$
Calculate residuals, $R = \{r_1, r_2, \ldots, r_o\}$
Calculate RMSE, ε using Equation (11)
Record ε value in $\varepsilon_{coll} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{(0.70)*N\text{-}o}\}$
Calculate relative error, e_{rel} using Equation (12)
Record e_{rel} value in $E_{rel\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.70)*N\text{-}o}\}$
Calculate absolute error, e_{abs} using Equation (13)
Record e_{abs} value in $E_{abs\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.70)*N\text{-}o}\}$
$m = m + o$ *// iterate*
end
Calculate the mean value of ε_{coll}, ε_{mean}
Calculate the mean value of $E_{rel\text{-}coll}$, $E_{rel\text{-}mean}$
Calculate the mean value of $E_{abs\text{-}coll}$, $E_{abs\text{-}mean}$
ε_{mean}, $E_{rel\text{-}mean}$, and $E_{abs\text{-}mean}$ = null
// testing
$m = 1$ *// variable to keep track of starting point of observations used in prediction*
// loop through all input values where $Y_{test} = \{y_1, y_2, \ldots, y_{(0.30)*N}\}$
for $j = o$ to (0.30)*N **do** *// N=size(Y) and iteration of o*
    Calculate predicted values using *predMean* retrieved from $predMean_{coll}$
    Calculate residuals, $R = \{r_1, r_2, \ldots, r_o\}$
    Calculate RMSE, ε using Equation (11)
    Record ε value in $\varepsilon_{coll} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{(0.30)*N\text{-}o}\}$
    Calculate relative error, e_{rel} using Equation (12)
    Record e_{rel} value in $E_{rel\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.30)*N\text{-}o}\}$
    Calculate absolute error, e_{abs} using Equation (13)
    Record e_{abs} value in $E_{abs\text{-}coll} = \{e_1, e_2, \ldots, e_{(0.30)*N\text{-}o}\}$
    $m = m + o$ *// iterate*
end
Calculate the mean value of ε_{coll}, ε_{mean}
Calculate the mean value of $E_{rel\text{-}coll}$, $E_{rel\text{-}mean}$
Calculate the mean value of $E_{abs\text{-}coll}$, $E_{abs\text{-}mean}$
ε_{mean}, $E_{rel\text{-}mean}$, and $E_{abs\text{-}mean}$ = null

Figure 5. Algorithm of threshold based experimental setup.

RESULTS

The results based on least squares, maximum likelihood estimation, segmentation averaging, and threshold-based statistical processing methods are presented for temperature, humidity, and energy usage sensors. Generated performance metrics are mean RMSE, relative error, and absolute error. The objective is to determine the number of observations, *o*, and statistical method that generates, in order of priority, the lowest RMSE, relative error, and absolute error.

Least Squares

The least square results for all window sizes, *o*=6 (1 ½ hours), 12 (3 hours), 24 (6 hours), 48 (1/2 day), and 96 (1 day) are shown in Table 1. Results from the training sets shows the lowest absolute error from *o*=6 for all sensor types. The trend from training illustrates error increasing as *o* increases. The size that gave the smallest absolute error for test cases is *o*=24 for humidity and energy and *o*=12 for temperature data.

Table 1. Data Prediction Accuracy as a Function of Window Size Using Least Squares

TRAIN									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	1.1695	0.0209	0.016	2.153	0.036	0.024	6.639	2.202	0.051
12	1.699	0.031	0.021	3.196	0.055	0.032	9.341	3.173	0.065
24	2.272	0.041	0.027	4.352	0.070	0.042	10.157	3.508	0.071
48	3.755	0.070	0.043	6.530	0.107	0.062	10.609	3.623	0.073
96	55.385	0.941	0.741	56.946	0.947	0.623	15.784	1.172	0.095
TEST									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	4.652	0.093	0.066	8.045	0.152	0.093	21.547	7.483	0.171
12	3.438	0.064	0.042	5.225	0.095	0.057	100.199	9.940	0.471
24	5.812	0.086	0.053	6.096	0.095	0.054	24.657	8.900	0.123
48	4.205	0.077	0.051	91.489	0.697	0.323	11.468	3.948	0.080
96	56.136	0.953	0.745	56.877	0.952	0.626	16.173	1.185	0.098

Maximum Likelihood Estimation

The maximum likelihood estimation results for all window sizes, *o*=6 (1 ½ hours), 12 (3 hours), 24 (6 hours), 48 (1/2 day), and 96 (1 day) are shown in Table 2. Results from the training sets shows the lowest absolute error from *o*=6 for temperature, humidity, and energy data. The trend from training illustrates error increasing as *o* increases. The size that gave the smallest absolute error for test cases is *o*=12 for temperature and humidity and *o*=96 for energy data. An interesting observation on testing for energy data is error increases as *o* decreases.

Table 2. Data Prediction Accuracy as a Function of Window Size Using Maximum Likelihood Estimation

TRAIN									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	1.824	0.034	0.023	4.229	0.072	0.043	9.953	3.523	0.072
12	2.183	0.039	0.026	4.334	0.073	0.043	10.193	3.605	0.072
24	2.839	0.048	0.033	5.683	0.092	0.054	10.48	3.726	0.074
48	3.899	0.063	0.045	7.748	0.126	0.074	10.627	3.732	0.074
96	4.862	0.079	0.056	9.733	0.161	0.093	10.691	3.764	0.074
TEST									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	2.398	0.047	0.031	4.553	0.083	0.049	11.104	4.234	0.084
12	2.494	0.046	0.031	4.707	0.082	0.048	11.188	4.117	0.081
24	3.012	0.052	0.036	6.029	0.101	0.059	10.815	3.859	0.077
48	3.927	0.063	0.045	8.014	0.131	0.076	10.796	3.965	0.076
96	4.893	0.079	0.056	9.951	0.168	0.096	10.924	3.919	0.076

Segmentation Averaging

The segmentation averaging results for all window sizes, *o*=6 (1 ½ hours), 12 (3 hours), 24 (6 hours), 48 (1/2 day), and 96 (1 day) are shown in Table 3. Results from the training sets shows the lowest absolute error from *o*=6 for humidity and energy and *o*=12 for temperature data. The size that gave the smallest absolute error for test cases is *o*=48 for temperature and *o*=6 for humidity and energy data.

Table 3. Data Prediction Accuracy as a Function of Window Size Using Segmentation Averaging

TRAIN									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	10.166	0.163	0.134	7.949	0.193	0.080	9.944	3.657	0.071
12	10.015	0.161	0.131	15.140	0.251	0.161	10.548	3.679	0.075
24	10.121	0.160	0.130	15.423	0.251	0.161	10.655	3.694	0.075
48	10.350	0.160	0.130	15.744	0.250	0.161	10.808	3.711	0.075
96	10.561	0.159	0.130	16.115	0.250	0.161	10.900	3.753	0.075
TEST									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	10.011	0.162	0.134	8.087	0.212	0.086	9.927	3.405	0.074
12	9.881	0.160	0.131	14.911	0.251	0.162	10.407	3.716	0.076
24	10.129	0.161	0.131	15.196	0.250	0.162	10.559	3.704	0.075

48	10.253	0.158	0.129	15.727	0.249	0.159	10.515	3.733	0.075
96	10.473	0.158	0.129	16.012	0.249	0.160	10.575	3.690	0.075

Threshold Based

The threshold based results are shown in Table 4 through 6 for c = 1, 2, and 3 respectively. Observation window sizes, o=6 (1 ½ hours), 12 (3 hours), 24 (6 hours), 48 (1/2 day), and 96 (1 day) are investigated. Results from the training sets shows the lowest absolute error from o=6 for temperature, humidity, and energy data when c=1, 2, and 3. The trend from training illustrates error increasing as o increases. The size that gave the smallest absolute error for test cases is o=6 for temperature, humidity, and energy when c=1 and 3. When c=2 for test cases, the absolute error is minimal when o=6 for temperature and humidity and o=12 for energy data.

Table 4. Data Prediction Accuracy as a Function of Window Size Using Threshold Based (c = 1)

TRAIN									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	2.017	0.039	0.025	4.081	0.072	0.042	11.099	2.400	0.073
12	2.325	0.042	0.028	4.919	0.082	0.048	11.257	2.540	0.075
24	3.021	0.051	0.035	6.451	0.102	0.062	11.371	2.591	0.075
48	4.134	0.063	0.046	8.883	0.136	0.085	11.420	2.618	0.075
96	5.137	0.080	0.058	10.852	0.169	0.105	11.329	2.658	0.075
TEST									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	1.944	0.039	0.026	3.931	0.073	0.042	10.186	2.413	0.073
12	2.264	0.042	0.028	4.684	0.081	0.048	10.632	2.620	0.074
24	3.008	0.052	0.036	6.308	0.102	0.062	10.941	2.603	0.074
48	4.134	0.065	0.047	8.703	0.135	0.085	11.218	2.596	0.075
96	5.074	0.079	0.057	10.759	0.171	0.104	11.724	2.538	0.077

Table 5. Data Prediction Accuracy as a Function of Window Size Using Threshold Based (c = 2)

TRAIN									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	1.995	0.038	0.025	3.746	0.067	0.038	10.194	3.671	0.074
12	2.266	0.041	0.028	4.421	0.075	0.044	10.461	3.742	0.074
24	2.889	0.050	0.034	5.788	0.095	0.055	10.546	3.695	0.074
48	3.887	0.062	0.044	7.818	0.127	0.074	10.635	3.734	0.075
96	4.852	0.079	0.056	9.760	0.162	0.093	10.723	3.762	0.075
TEST									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*

6	1.918	0.038	0.025	3.618	0.067	0.039	9.707	3.594	0.073
12	2.211	0.041	0.028	4.288	0.075	0.044	9.986	3.553	0.073
24	2.822	0.049	0.034	5.585	0.093	0.055	10.549	3.756	0.075
48	3.907	0.063	0.045	7.494	0.128	0.074	10.660	3.724	0.075
96	4.874	0.079	0.056	9.820	0.166	0.095	10.772	3.698	0.075

Table 6. Results of Data Prediction using TB (c = 3)

TRAIN									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	1.994	0.038	0.025	3.746	0.066	0.038	10.139	3.706	0.073
12	2.253	0.041	0.028	4.433	0.075	0.044	10.367	3.724	0.074
24	2.884	0.049	0.034	5.783	0.094	0.055	10.536	3.696	0.074
48	3.908	0.063	0.045	7.827	0.127	0.074	10.668	3.796	0.075
96	4.849	0.079	0.056	9.763	0.163	0.099	10.753	3.740	0.075
TEST									
	Temperature (°F)			Humidity (%RH)			Energy (Wh)		
o	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*	*RMSE*	*RE*	*AE*
6	1.934	0.038	0.025	3.642	0.067	0.039	9.827	3.699	0.074
12	2.247	0.042	0.028	4.288	0.075	0.044	10.152	3.711	0.074
24	2.814	0.049	0.034	5.599	0.094	0.055	10.424	3.814	0.075
48	3.849	0.062	0.044	7.732	0.127	0.075	10.604	3.626	0.075
96	4.867	0.079	0.056	9.821	0.164	0.095	10.671	3.803	0.075

CONCLUSION

Autonomous data correction for building data is studied using statistical processing methods, namely least squares, maximum likelihood estimation, segmentation averaging, and threshold based. This is accomplished by using observation windows which define a subset of samples, size *o*, that are used to generate a model. Validation and correction occurs for each successive observation window within the sensor dataset using interpolation and/or extrapolation for missing and corrupt data. Tables 7 through 9 summarize the best performing cases for temperature, humidity, and energy data respectively. The threshold based technique performed best with temperature (c=2), humidity (c=2), and energy data (c=1).

While it is anticipated that the temperature, relative humidity, and energy used in this study would follow similar patterns in other buildings, it should be noted that the sensor data used in this study came from one building and that additional study would be needed to confirm the degree to which these results generalize across the building stock. This includes future research using temperature, relative humidity, and energy data with various profiles.

Other future work in autonomous data correction for building data is studying other types of methods besides statistical such as filtering and machine learning techniques. Other data types will also be investigated such as heat flux, airflow, and liquid flow. The study of dependent data prediction which uses other sensor data will also be considered. This includes using ambient

temperature and humidity, home occupancy, time of the day, day of the week, etc… Comparison of automated methodologies to corrupted or missing sensor data corrected by domain experts is planned to validate the utility of these approaches.

Lessons learned from these studies can be used to develop software tools for data visualization and analytics, ensuring data validity and improved understanding. This would be particularly useful for large datasets where manual validation would be next to impossible. Focus can also be shifted from software to hardware, implementing these techniques on a hardware platform, for real-time data validation at the sensor node.

Table 7. Results of Best Performers for Temperature Data

	Temperature (°F)							
	TRAIN				**TEST**			
	o	*RMSE*	*RE*	*AE*	*o*	*RMSE*	*RE*	*AE*
LS	6	1.1695	0.0209	0.016	12	3.438	0.064	0.042
MLE	6	1.824	0.034	0.023	12	2.494	0.046	0.031
SA	12	10.015	0.161	0.131	48	10.253	0.158	0.129
TB (*c* = 1)	6	2.017	0.039	0.025	6	1.944	0.039	0.026
TB (*c* = 2)	6	1.995	0.038	0.025	6	1.918	0.038	0.025
TB (*c* = 3)	6	1.994	0.038	0.025	6	1.934	0.038	0.025

Table 8. Results of Best Performers for Humidity Data

	Humidity (%RH)							
	TRAIN				**TEST**			
	o	*RMSE*	*RE*	*AE*	o	*RMSE*	*RE*	*AE*
LS	6	2.153	0.036	0.024	24	6.096	0.095	0.054
MLE	6	4.229	0.072	0.043	12	4.707	0.082	0.048
SA	6	7.949	0.193	0.080	6	8.087	0.212	0.086
TB (*c* = 1)	6	4.081	0.072	0.042	6	3.931	0.073	0.042
TB (*c* = 2)	6	3.746	0.067	0.038	6	3.618	0.067	0.039
TB (*c* = 3)	6	3.746	0.066	0.038	6	3.642	0.067	0.039

Table 9. Results of Best Performers for Energy Data

	Energy (Wh)							
	TRAIN				**TEST**			
	o	*RMSE*	*RE*	*AE*	o	*RMSE*	*RE*	*AE*
LS	6	6.639	2.202	0.051	24	24.657	8.900	0.123
MLE	6	9.953	3.523	0.072	96	10.924	3.919	0.076
SA	6	9.944	3.657	0.071	6	9.927	3.405	0.074
TB (*c* = 1)	6	11.099	2.400	0.073	6	10.186	2.413	0.073
TB (*c* = 2)	6	10.194	3.671	0.074	12	9.986	3.553	0.073
TB (*c* = 3)	6	10.139	3.706	0.073	6	9.827	3.699	0.074

ACKNOWLEDGMENTS

Research sponsored by the Laboratory Directed Research and Development Program (WN12-036) of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy (DE-AC05-00OR22725).

REFERENCES

Bo, A., Zhang-Dui, Z., Gang, Z., and Jian-Ping, L. (2009). Novel statistical processing methods for wireless field strength prediction. *IEEE Transactions on Consumer Electronics*. vol. 55. no. 4. pp. 1805-1809.

Christian, J. (2010). Comparison of retrofit, advanced and standard builder's homes in Campbell Creek. *Thermal Performance of the Exterior Envelopes of Buildings, IX, Proceedings of ASHRAE THERM XI*. Clearwater, FL.

Frolik, J., Abdelrahman, M., and Kandasamy, P. (2001). A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data. *IEEE Transactions on Instrumentation and Measurement*. vol. 50. no. 6. pp. 1761-1769.

Harris J.W. and Stocker, H. (1998). *Handbook of mathematics and computational science*. Springer-Verlag. New York, New York.

Hoel, P.G. *Introduction to mathematical statistics.* (1984). 5th edition. John Wiley & Sons.

Ibarguengoytia, P.H., Sucar, L.E., and Vadera, S. (2001). Real time intelligent sensor validation. *IEEE Transactions on Power Systems.* vol. 16. no. 4. pp. 770-775.

Intergovernmental Panel on Climate Change. (2007). *Summary for policymakers. In climate change 2007: mitigation. Contribution of working group III to the fourth assessment report of the intergovernmental panel on climate change*. Cambridge University Press. Cambridge, United Kingdom and New York. Metz, B., Davidson, O.R., Bosch, P.R., Dave, R., and Meyer, L.A. (Eds).

Miller W.A. and Kosny, J. (2008). Next generation roofs and attics for homes. *ACEEE Summer Study on Energy Efficiency in Buildings, Proceedings of American Council for an Energy Efficient Economy*, Pacific Grove, CA.

Miller, W., Kosny, J., Shrestha, S., Christian, J., Karagiozis, A., Kohler, C., and Dinse, D. (2011). Advanced residential envelopes for two pair of energy-saver homes. Retrieved from http://www.zebralliance.com/pdfs/ZEBRA%20Demonstration%20Report.pdf URL.

Norton, P. and Christensen, C. (2006). A cold-climate case study for affordable zero energy homes. *Proceedings of Solar 2006, American Solar Energy Society*. Denver, CO.

Parker, D.S., Sonne, J.K., Sherwin, J.R., and Moyer, N. (2011). Comparative evaluation of the impact of roofing systems on residential cooling energy demand in Florida. Final Report FSEC-CR-1220-00, prepared for Florida Power and Light Company.

Postolache, O.A., Girao, P.M.B.S., Pereira, J.M.D., and Ramos, H.M.G. (2005). Self-organizing maps application in a remote water quality monitoring system. *IEEE Transactions on Instrumentation and Measurement*. vol. 54. no. 1. pp. 322-329.

U.S. Department of Energy. (2008). Buildings energy data book. Retrieved from http://buildingsdatabook.eren.doe.gov/ URL.

U.S. Green Building Council Research Committee. (2007). A national green building research agenda. Retrieved from http://www.usgbc.org/ShowFile.aspx?DocumentID =3102 URL.

Uluyol, O., Kim, K., and Nwadiogbu, E.O. (2006). Synergistic use of soft computing technologies for fault detection in gas turbine engines. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*. vol. 36. no. 4. pp. 476-484.

ZEBRAlliance. (2008). ZEBRAlliance: building smart. Retrieved from http://www.zebralliance.com/index.shtml URL.

芽|Sprouts

Working Papers on Information Systems | ISSN 1535-6078

Office:
Sprouts
University of Amsterdam
Roetersstraat 11, Room E 2.74
1018 WB Amsterdam, Netherlands
Email: admin@sprouts.aisnet.org

芽|Sprouts