Smart Sensing for Building Monitoring and Energy Management Applications

ABSTRACT

Energy-aware consumption can significantly improve building performance. But to make people energy-aware, information regarding energy minimization opportunities needs to be available at the right time and in the right format. Wireless sensor networks (WSNs) have found many applications in this regard for gathering information relevant for building environment monitoring and energy management. However, sensor data is not completely reliable and hence its use in energy management has to be carefully considered. Experiments conducted on a WSN test bed monitoring a campus classroom highlight how sensor data is prone to errors, underlining the need to detect faulty measurements from the data. This paper describes the WSN test bed, the experiments showcasing faulty measurements as well as methods used to detect these faulty measurements. A key contribution of this paper is highly accurate fault detection methods developed using two machine learning approaches: neural networks and support vector machine. The paper also provides insights on implementation of the proposed methods as well as its integration into an energy management platform. Finally, example studies demonstrating the use of sensor data for detecting energy wastage and optimizing energy use are described.

CCS CONCEPTS

•Hardware →Error detection and error correction; Sensor applications and deployments; Temperature monitoring; **•Computing methodologies →***Support vector machines; Neural networks;*

KEYWORDS

wireless sensor networks, sensor fault detection, energy monitoring

ACM Reference format:

. 2017. Smart Sensing for Building Monitoring and Energy Management Applications. In *Proceedings of ACM eEnergy conference, HONG KONG, May 2017 (ACM eEnergy),* 12 pages.

DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Smart consumers are slated to form a key part of the supply-demand balance equation of future power grids. They are expected to be not only energy efficient, but also dynamically adapt their power consumption to suit the needs of the power grid. The last decade has seen a significant amount of research and pilot programs across the world geared towards leveraging demand-side flexibility in power system operations. Of particular note are the GridWise demonstration projects spearheaded by Pacific Northwest National Laboratory [7], which emphasize how interactive display of information regarding their electric consumption can enable consumers to make effective decisions that benefit the grid. Faruqui et al have also concluded that information from the utility can go a long way in eliciting an appropriate response from the consumers [4]. Likewise, it is equally important for consumers to be informed about their own loads and the environmental conditions which govern them [15].

This paper describes on-going research wherein a wireless sensor network (WSN) has been deployed at a university campus building to gather the required data regarding the consumer's environment. An interactive platform has been developed to display this data along with alerts for optimizing the energy consumption. But monitoring without processing data does not have any significance since sensors are always noisy and sometimes could lead to wrong control decision. Hence, two fault detection methods have been developed based on neural networks and support vector machine. The paper documents their development and demonstrates their performance on the real world WSN test bed data set.

1.1 Motivation

The use of sensors for minimizing energy wastage is a well know concept. For many years, many commercial buildings have been equipped with passive infrared (PIR) sensors in corridors/restrooms to detect occupants and suitably control lighting. Now, with variety of sensors available at affordable costs, they are finding many applications in energy management applications. A comprehensive survey of how sensors are being used to predict occupancy and hence optimize energy consumption in residential and commercial buildings is available in [11]. Furthermore, as wireless technologies mature and become more economical, WSNs are expected to play a crucial role in future building energy management systems. It is touted that the information gleaned from the WSNs can be effectively used in improving energy efficiency, lowering electricity bills and enhancing demand responsiveness [1, 9]. For such practical applications remote energy monitoring is required which involves different sensors to sense environment. For energy monitoring application to be reliable it is important that integrity of the sensor data is not compromised. This motivates us to develop fault detection methods using machine learning approaches.

1.2 Related Work

Fault detection using of sensor data is very important to avoid any kind of system failures and retain reliability of a control system. Georg Jager et. al[8] have shown results in using neural networks to detect faults in the sensor data. However accuracy that is obtained in their data is not reliable for practical applications. Xiao Xu et.

ACM eEnergy, HONG KONG

DOI: 10.1145/nnnnnnn.nnnnnnn

al.[14] demonstrates another approach for fault detection in sensor data by modelling drift between estimation and measured values of sensor. F. Koushanfar et. al.[10] have demonstrated on-line model based testing technique that can be generalized to arbitrary system of heterogeneous sensors with an arbitrary type of fault. However their limitation is they require sufficient number of non-faulty sensors to accurately specify impact on the environment. Yang Zhang et. al.[16] have discussed various methodologies used for outlier detection in wireless sensor networks data. Pardis, Lilia et. al. [13] have surveyed different techniques used to address problem of fault detection in wireless sensor networks in several application areas. They further discuss algorithms to prevent, detect, identify and treat faults. This research is mainly aimed at developing fault detection model based on single sensor data alone and then try to generalize this model to other similar technology sensors.

1.3 Contributions of the Paper

The main contribution of this paper is development of fault detection methods based on neural networks and support vector machines, which leverage the statistical features of a single sensor data to identify faults. The proposed methods have been deployed on the actual WSN test bed data and they have shown a remarkably high accuracy in fault detection. Furthermore, a WSN-based software platform for energy management that displays energy usage related information in an intuitive manner is developed. The platform adopts smart practices for bad data detection and correction. The information available from the WSN can be effectively used to avoid energy wastage and optimize end-use consumption. While the platform can be implemented for any building type – be it residential, commercial or industry – we explain the approach in the context of the experimental set up available at the university campus. In particular, the WSN has been deployed in an exemplary classroom to monitor its internal and external environment. We use actual recorded instances from this classroom to demonstrate the capabilities of our energy management platform.

This paper is organized as follows: Section 2 describes the experimental setup inside the classroom in terms of the setting up of the WSN, wireless energy meters and other components. We also discuss analysis of the data obtained from WSN sensors. We also discusses techniques for sensing and bad data detection and correction leveraging correlation among group of sensors. In section 3 we discusses different types of faults that can be observed and types of faults observed in test bed data. We also discuss about the methodology used for fault detection in wireless sensor network. In section4 we discuss the results obtained in developing classification model for fault detection. Finally, Section 5 describes the software platform to manage energy consumption in an efficient manner. We conclude with a discussion on ongoing work and future work.

2 EXPERIMENTAL SETUP

We describe in this section the experimental setup at the university campus. A brief overview of the classroom, the WSN, the wireless energy meters and data collection process is provided below.

A 150-seater classroom located on the topmost floor of the lecture hall complex on the university campus is selected for our experiments since it presents a worse case scenario in terms of air conditioning load: least shade available and heat influx from the roof. The classroom is equipped with variable refrigerant flow (VRF) type air-conditioner controlled remotely from an office located on the same floor. The VRF system consists of an outdoor unit to supply cool air and 8 indoor units to circulate this cooled air, as seen in Fig. 1. The outdoor unit is shared with another classroom located just below the classroom chosen for experiments.

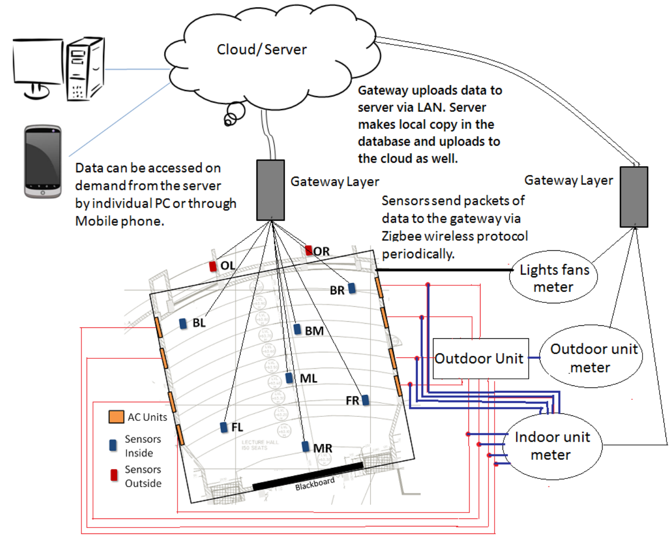


Figure 1: The experimental setup in the classroom located in the university campus.

A total of 66 tube lights and 21 CFLs are used for lighting the entire classroom. Ten fans are also operated in the classroom in uniformly distributed locations. In addition, the classroom also operates two projectors, a computer and two wifi routers whose contribution to the energy consumption is small compared to lights, fans and air-conditioning.

The classroom environment is monitored using temperature, humidity and luminosity sensors. The WSN used for monitoring consists of battery-powered nine sensor motes – two of which are placed outside and the rest are strategically spread out across the classroom – as well as a coordinator. All nine sensor motes communicate to the gateway in star topology. The position of the nine sensor motes is shown in Fig.1.

Each mote (Waspmote from Libelium) houses sensors for temperature, luminosity and humidity. Data from these sensors are sampled every 5 minutes. ZigBee protocol based XBee-pro radio module from Digi international provides wireless communication between sensor motes and the gateway. Linux-based Beagle bone black minicomputer is used as a bridge between ZigBee protocol and Internet protocol. The server is housed by Dell Poweredge T20 machine that runs Linux OS. The data flow on the network is shown in Fig.1.

2.1 Spatial Correlation Within Sensors

Error detection and correction for the sensor data is critical when we use this data for energy optimization. While on-board signal conditioning circuitry can filter out errors in sensor data, it leads to more power consumption. We leverage the spatial correlation observed within different sensor motes and use this to devise effective bad data detection schemes. The specific schemes are detailed here.

Since the spatial spread of the sensors is not large, the sensor readings may exhibit high correlation. Sensors with high correlation can be grouped together to exploit the following benefits:

- Predicting values of other sensors in a group from one sensor
- Increasing life of sensor by using smart sampling techniques
- Bad data detection and correction

Since wireless communication can occasionally incur loss of data packets, the capability to predict sensor readings can be leveraged to make up for the missing data from other available data. For instance, regression analysis on temperature sensors of FL, FR, ML and MR motes leads to the following relationship

$$FR = -2.79 + 0.21FL + 0.65MR + 0.22ML \,. \tag{1}$$

This relationship may be tuned to seasonal changes when correlation worsens. When the readings obtained from (1) were compared with the actual readings from FR temperature sensor, the error could be fit to a Gaussian distribution with mean of 0.0069^{o}C and standard deviation of 0.3^{o}C. The maximum error on the validation data set was 1^{o}C, roughly 5% of the observed value, which does not have any significant impact on the energy optimization problem discussed in the further sections.

2.2 Bad Data Detection and Correction

Since faulty sensor readings can impact consumer decisions regarding optimizing energy usage, bad data detection and correction is crucial. We leverage the inherent redundancy in the WSN as well as the spatial correlation between sensors to identify faulty sensor readings and correct them[5]. Furthermore, historical data can be used to set up simple checks based on minimum and maximum expected readings which can also be incorporated for bad data detection.

The maximum rate of change in temperature and humidity observed across subsequent readings is bounded by slow environmental variations and maximum BTU capacity of AC. Since high frequency errors can occur due to inherent noise in the sensor, a suitable bound may be adopted to identify such erroneous data. Finally, knowledge regarding location of sensors also provides significant insights to guide error detection. For instance, sensors near the indoor AC outlets would be expected to have lower temperatures than those away from the outlets. Such spatial relationships can also be used to detect faulty measurements.

To summarize, a measurement is classified as faulty if it disobeys one of the following:

- Maximum and minimum bounds
- Maximum rate of change
- Spatial relation with other sensors
- Relationship established via regression analysis

The above checks are applied to detect bad data from the FR temperature sensor. Fig.2 shows measurements from FL and FR temperature sensors along with the errors that have been flagged. Since the temperatures recorded by FL and FR are highly correlated, the FR temperature measurements are expected to follow the trends seen in FL temperature measurements. The red points are incorrect measurements recorded by FR sensor that have been flagged. Note that the errors within prediction error bound of 1^{o}C remain undetected. However, inherent control characteristics of the VRF AC system often result in fluctuations of 1^{o}C from set temperature. Hence, errors undetected within $\pm 1^{o}$C range do not adversely impact the decision making model and are thus not our concern.

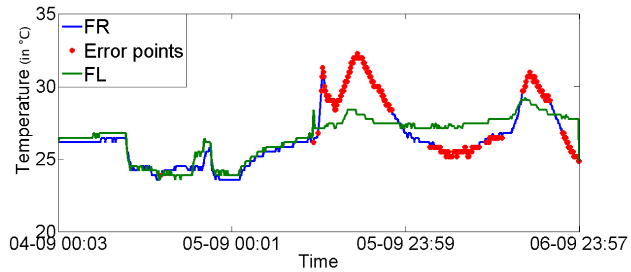


Figure 2: Detection of incorrect temperature readings from FR sensor, with errors flagged in red.

2.3 Data Analysis

Data from wireless sensor nodes was observed to study patterns of temperature and humidity variations inside the classroom. Furthermore, these variations were also studied to examine the relationships with energy consumption patterns. This analysis has helped us to identify over usage of energy in some instances from the observed data. Following sections discuss analysis of the observed data and simple rule based fault detection method.

Temperature data from all the sensors were studied to observe relationships among the sensors. Some sensors were observed to be highly correlated. Data for correlation was observed for the period of 20th Aug to 1st Sept 2015. Correlations higher than 0.94 are considered to form group of sensors. This value is chosen because this will give about 5-6% prediction error during regression analysis of these measurements. This error margin is tolerable as maximum error in temperature is 1°C only, which is same as maximum amount of fluctuations observed in temperature data from set point of AC. We form group of sensors that have high correlation coefficient. This has following advantages,

- Increasing lifetime of the network [2]: Correlation can be leveraged to increase lifetime of the network by sampling from a single sensor in a correlation group and constructing values of other sensors from available single value. This way for N number of sensors in correlated group life of the network becomes N times the life using single sensor. Drawback of this approach are as follows,
  - More number of sensor are required to be installed.
  - This adds redundancy to the system and increases its net cost.
- Detecting faulty measurements: Values of the current sensor are compared with other sensors to get sense of correlation pattern and faults in the system can be detected. Drawback of this method are as follows,

  - It becomes difficult to analyze if more sensors have faulty measurements.
  - Correlation of the sensors changes over time so shall the comparison metric.
- Prediction and relearning of regression coefficients: As correlation of the sensors changes over time, it can be re-evaluated and regression coefficients can be updated automatically.

Humidity data relationships among sensors can be leveraged to detect faults that might occur in any one of the sensors. Similar correlation coefficient values are observed between same group of sensors for humidity and temperature sensors. For such high correlation values relationship can be obtained by using simple linear regression on the data from sensors.

Sensors motes deployed do not have an electronic hardware for signal conditioning. This is to have minimum hardware on sensor boards and hence minimize board power consumption. But this comes at cost of errors in sampled data. Such errors in measurements can cause incorrect decisions. Incorrect decisions can cause serious failures in control systems we intend to develop to autonomously tune the environmental conditions. Figure 2 shows an instance of an erroneous data. The above mentioned data analysis strategy can help in detecting and correcting such errors.

One common drawback of wireless sensors is low reliability. Due to this some sensors fail to transmit frames or controller misses the transmitted frames. This can be easily avoided by re-transmitting a frame upon no acknowledgement from controller but transmission step consumes more power. So instead of re-transmitting a frame, that frame can reconstructed using interpolation and validated with prediction from correlated sensors. This is an overhead only on the server and hence can be easily implemented without draining power from sensors thus not affecting wireless sensor network life.

2.4 Relationships Among Sensors

As discussed earlier, some sensors exhibit high correlation coefficient value. This can be leveraged to form relationships between some sensors using regression analysis. These relationships can be used to predict values of one sensor from another in case faults detected in the sampled data. Following relationships were observed by obtaining coefficients using method of normal equations and assuming linear relationships among temperature sensors. Linear relationship can be assumed due to high value of correlation coefficient. Groups that are formed are as follows,

- Group 1 : FR, FL, MR, ML
- Group 2 : BL, BR
- Group 3 : OL, OR

Largest observed group is group 1 with 5 sensors correlated with coefficient greater than 0.94. The specific relations are obtained using linear regression of the kind presented in (1).

2.5 Fault Diagnostics

Based on the analysis discussed in the previous sections, certain rules can be formed to detect faults in the observed data. Such faults can very well be corrected using relationships described earlier. Following is validation algorithm to detect faults in signal,

- Check upper and lower bounds on measurement
- Check rate of change in temperature of measurement
- For inside sensors, for AC ON period temperature should be within 1°C bound from set temperature
- Check spatial relationship among sensors
- Check prediction error for sensor, mark errors if prediction error is more than range obtained during regression

For a group of correlated sensors, values predicted from other sensors can be used as measurement value. Even then corrected value can have error and that is within bound given by prediction error during learning. Figure2 shows results using above mentioned validation technique. It clear from figure that invalid points marked red can be identified using this method. Some points that are not detected even though they follow pattern of error as they are within prediction error, this issue is addressed in next chapter by modelling fault detection problem in a different way.

3 SENSOR FAULT DETECTION

While correlation can be leveraged for sensor fault detection (as explained in the preceding section), it is not always viable to adopt this approach, particularly when sensor data is being use for energy management application as in the case of this research. The reasons for this are many fold:

- Temperature and humidity patterns change over time so does correlation,
- Implementation requires redundancy in sensor nodes which may not be feasible due to budget concerns,
- Correlation-based methods may not work if more number of sensors in a correlated group are faulty, and,
- Rule-based algorithms will perform poorly when there are sudden climatic changes.

These drawbacks are result of pure and unprocessed data based rules formed by the algorithm and dependency on correlated sensors.

Statistics of the data provides richer information than just pure data alone. Statistical parameters do vary with the data and depict more patterns in the data than pure data alone. So, building a fault detection algorithm based on statistical patterns of the measured data may prove to be more reliable than pure and unprocessed data-based rule model. Correlation dependency is also eliminated, redundancy is unnecessary and the statistics will capture underlying parameters such as location of sensor node and climate changes. Hence, faults can be detected reliably and model can be easily generalized to multiple sensors. This section describes the algorithms adopted for statistics-based fault detection. The following subsections describes the type of faults seen in the test bed data, feature design for identifying these faults and methods used in the identification process.

3.1 Types of Faults

Faults occur in both the domains time and amplitude [12, 17]. For instance, measurements may be recorded with delay as compared to their actual/true value or they may be dropped – these are faults in the time domain. Alternately, faults may affect the amplitude of the data measured by the sensors as follows:

- Offset: Measurements differ from the true values by a constant or time-varying offset.

- Stuck-at-X: Measurements relay the same value for a long time interval.
- Outlier: Measured values abruptly transition towards a large value which is normally not seen by the system.
- Spike: As compared to an outliers, measurements experience a smoother transition to a high value – different from the true value.

A more detailed modeling of faults is discussed in[12, 17]. The discussion of results explains the type of faults seen in the recorded test bed sensor data. A point to note is that time domain faults are harder to characterize and are not considered in this research. The following subsection discusses statistical features that may be employed to characterize amplitude domain faults commonly seen in the sensor data from the WSN test bed.

3.2 Feature Extraction

Statistical information contained in sensor measurement data can be leveraged to identify faulty measurements. The critical question is which statistical features to use? Clearly, the nature of faults expected to be seen in the data set plays a vital role in feature extraction. Furthermore, these features should be non-redundant so that the fault detection algorithm converges faster and is computationally tractable. In what follows, the specific features used for fault detection deployed at the WSN test bed are described.

Since the main focus of this research is on amplitude domain faults, the following statistical parameters based on a moving window of T samples are used in detection algorithms:

- Mean: It is calculated as the average of recent T measurements. If $z(t)$ is the t^{th} measurement, then the corresponding mean $\mu(t)$ is expressed as

$$\mu(t) = \frac{1}{T} * \sum_{\tau=0}^{T-1} z(t-\tau) \tag{2}$$

  Mean helps in classifying an outlier and stuck at X faults, as in such instances mean value will be higher than usual.
- Standard deviation: At instance t, it is denoted by $\sigma(t)$ and calculated using the T recent measurements as follows

$$\sigma(t) = \frac{1}{T} * \sqrt{\sum_{\tau=0}^{T-1} (x(t-\tau) - mu(t))^2} \tag{3}$$

  Standard deviation quantifies the width of the measurements' distribution and, thus, quantifies the range of possible true measurements.
- Signal to noise ratio: At instant t, it is denoted by $\mathrm{SNR}(t)$ and calculated as the reciprocal of the coefficient of variation as follows

$$\mathrm{SNR}(t) = \frac{\mu(t)}{\sigma(t)} \tag{4}$$

  It is simply ratio of corresponding average value and standard deviation. In case of faults, standard deviation is high and SNR value is low but during true measurements SNR is high as standard deviation is very low. Thus, it is a useful characteristic for classifying faults.
- Maximum deviation: It is the maximum deviation observed in the recent T measurements. It is denoted by $\mathrm{MD}(t)$ for the t^{th} measurement and calculated as

$$\mathrm{MD}(t) = \max\{|z(t-\tau) - \mu(t)|\ \tau = 0, 1, \ldots, T-1\} \tag{5}$$

  This feature will take a high value, even for small spikes/outliers which may otherwise go undetected and is hence useful in fault detection.
- Velocity: It is instantaneous rate of change in sensor measurement. If Δt is the sampling period, then velocity $v(t)$ at instance t is

$$v(t) = \frac{z(t) - z(t-1)}{\Delta t} \tag{6}$$

  Velocity quantifies sudden changes in the measurements and is hence useful quickly classify high frequency fluctuations in the sensor data.
- Measurement value: Finally, the measurement value at each instance $z(t)$ is in itself a useful feature for fault detection.

A rigorous reasoning is used to extract six features that are deemed to be informative about the measurement data. Principle component analysis (PCA) [6] is performed on these features to identify redundancy, if any. Figure 3 shows component values calculated on training data used in the results section. It is evident from the figure that no component has null value i.e. zero standard deviation and hence feature set has independent feature vectors.

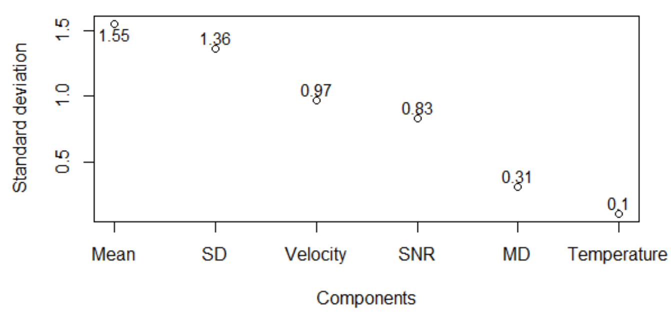


Figure 3: PCA plot for feature set with window size $T = 5$

3.3 Fault Detection Methods

The key idea behind fault detection is to use the previously identified features to determine if measurement $z(t)$ at instance t is faulty or not. The input to the fault detection method would be six features (compactly represented by vector $X(t)$) calculated at instance t while its output would be a categorical variable $y(t)$ predicting whether fault exists or not. Several classification techniques such as *neural networks* (NN), *support vector machine* (SVM) and *Bayesian learning* may be deployed for mapping the feature set $X(t)$ to the response $y(t)$. The research described here used NN and SVM for fault detection.

Neural Network. The inspiration behind this method is the network of neurons: the idea is to use training data to train a representative network for fault detection. The network consists of interconnected cascade multiple layers of neurons. Interconnections between layers are assigned some values know as weights. During training, NN changes its weights so that most of the instances are classified correctly [6]. Since all the weights are in the forward directions such network is also know as feed forward neural network (FFNN). FFNN may be trained using backpropogation algorithm.

Support Vector Machine. SVM is an extension of support vector classifier which expands the feature space using special types of functions known as kernels. Unlike NN, SVM learns classifying plane and provides some margin or slackness along classifying plane. Objective of the SVM is to maximize margin along the classifying plane while minimizing error at the same time [6]; it thus guarantees global optimum for convergence on training data. Just as NN learns any non-linear hyperplane by adding more hidden layers and neurons, SVM learns non-linear hyperplanes using kernels. In SVM, the kernel is a function that quantifies the similarity between two observations. The kernel may be a linear function, a radial bias function (RBF) or a polynomial [6].

4 FAULT DETECTION IN TEST BED DATA

The temperature measurements collected from the WSN test bed were used to generate appropriate data sets for training and validating the fault detection approaches described in the preceding section. This section summarizes the results. First, an overview of data set generation is provided followed by the use of NN and SVM for fault detection. A comparison of the two methods and their applicability to generic settings is also presented.

4.1 Data Set Generation

The results reported herein use the temperature data samples collected in the WSN test bed. Feature set was formed with observations from 79292 data samples out of which 70% data was used for training purpose and 30% data was used for cross-validation purpose. As discussed previously, each and every measurement sample was *apriori* labeled as faulty or not by leveraging the spatial correlation amongst the test bed sensors. For test data set generation, three different months, wherein no fault was observed in the entire data, are selected. Using a computer program, the measurement data from these months are injected with faults representative of the faults observed in the FR sensor raw data. The three types of faults observed in the raw FR sensor data are outliers, offset and random noise (RN), as shown in figures 4 and 5.

As shown in fig. 4, FR sensor data has measurements close to 150°C instantaneously which is clearly a case of outlier. In figure4 we observe random noise causing signal to fluctuate at high frequency also in fig.5 FR sensor data was deviating from its normal behaviour as its measurements were drifting way far from AC set temperature which is clear case of offset with some noise. Such faults were randomly injected but different types of faults were injected at almost same number of times to maintain low bias in the test dataset. The following procedure was followed for generating test data:

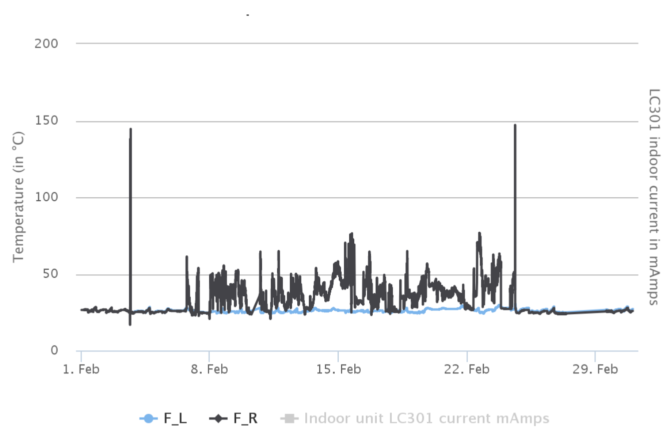


Figure 4: Outlier and random noise observed in the data

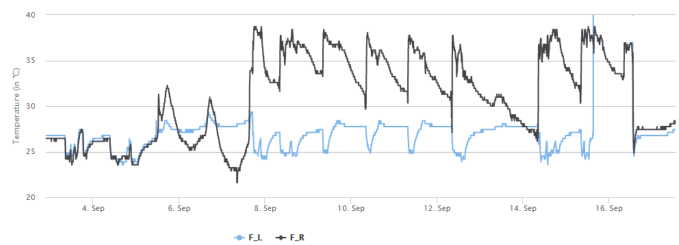


Figure 5: Offset observed in the data

- Sample data from a sensor consisting of only true values
- Add samples with outliers having value of 150 + Gaussian-noise(0,1)
- Add Gaussian random noise with mean *zero* and variance 4 to the existing true measurements
- Add offset having value of 4 to some existing measurements

Test dataset consists of 23311 instances out of which 19519(83.73%) are true measurements (no faults) and 3792(16.27%) are fault injected measurements. If an example is classified/labeled as TRUE then it is considered as a non-faulty measurement and if classified/labeled as FALSE then it is considered as faulty measurement.

4.2 NN-based Fault Detection

The FFNN for fault detection is formed with following configuration

- 6 input layer neurons with 1 bias unit - The six input neurons correspond to six components of the feature set.
- 4 hidden layer neurons with 1 bias unit - Different number of hidden layer neurons were tried out, but no significant change was observed in the performance beyond 4 neurons with one extra bias unit.
- 2 output layer neurons - The two output neurons represent two labels TRUE and FALSE, with either of them outputting "1" depending on whether the sample is classified as faulty or not.

This network was trained with backpropogation algorithm, with each neuron having sigmoid activation function and cross entropy function as an objective function. Separate dataset were used for training and testing a neural network as discussed previously. Table 1 shows confusion matrices obtained from the performance of neural network on cross-validation dataset with time window size of 3, 5 and 10 samples.

Table 1: Confusion matrix for window size $T = 3$, 5 and 10 on the cross-validation dataset using FFNN

Window size	Actual label	Classified TRUE	Classified FALSE
3	TRUE	20926	46
	FALSE	461	2357
5	TRUE	20953	6
	FALSE	550	2280
10	TRUE	20890	16
	FALSE	508	2374

With change in window size no significant change in performance of the network was observed. This is clearly evident from table 2 which shows that the true positive rate and F1-score is quite high independent of the window size used for feature set calculation. It also indicates that the false positive rate is also quite consistent and one possible reason could be that model is not able to detect all types of faults equally.

Table 2: Accuracy for different time window size using FFNN

Window size	TP rate	FP rate	F1 score
3	0.998	0.164	0.988
5	1.000	0.194	0.987
10	0.999	0.176	0.988

We performed multiclass classification using neural network to check if network poorly classifying some types of faults. Table 3 shows confusion matrix of multiclass classification of the data.

Table 3: Confusion matrix of multiclass classification with time window size 3, 5 and 10 samples on cross-validation dataset using FFNN

Window size	Actual label	Classified TRUE	Classified Offset	Classified Outlier	Classified RN
3	TRUE	20974	9	2	6
	Offset	309	606	0	27
	Outlier	0	1	728	4
	RN	155	120	3	836
5	TRUE	20992	0	2	4
	Offset	363	534	0	40
	Outlier	0	1	716	1
	RN	156	52	2	926
10	TRUE	20912	8	3	9
	Offset	329	613	0	38
	Outlier	1	0	749	3
	RN	136	70	3	914

It is clear from these results that all the outliers are correctly classified with good accuracy but classification is poor for offset and random noise classes. Before concluding that model is performing poorly on offset and RN classes we tried testing accuracy by adding some tolerance margin to error. Tolerance margin of 0.5°C is chosen for further evaluation for the same reason as described in section 3.7. This neural network was tested with test dataset and confusion matrix obtained is shown in table 4. Exactly 22900(98.23%) instances were correctly classified and 411(1.76%) instances were misclassified. Fig. 6 shows the data plot, with red points signifying misclassified points. Out of 411 misclassified points 263 points were close to true data by 0.5°C, which is a tolerable limit on temperature considering it is tolerance of sensing device is more. So only 148 points have been misclassified which leaves with error rate of 148/23311 = 0.634%. This suggests model will be wrong about 7 times for about 1000 instances observed. Fig. 7 shows histogram of the difference between values of misclassified point and true data, about 64% of the points lie within the error band of 0.5°C.

Table 4: Confusion matrix of test data with time window size 5 of FR sensor on test dataset using FFNN

Actual	Classified TRUE	Classified FALSE
TRUE	19456	63
FALSE	348	3444

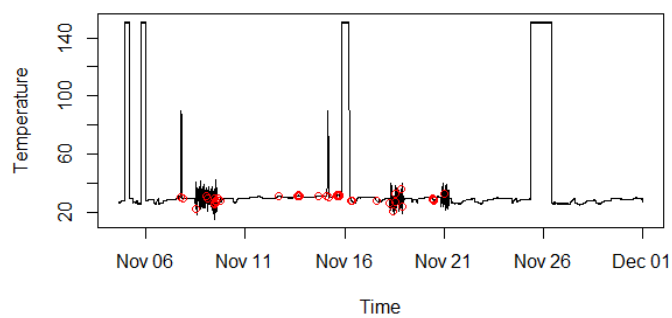


Figure 6: Prediction of model on the data for Nov 2015 using FFNN

4.3 SVM-based Fault Detection

SVM is chosen for comparing performance with FFNN as SVM performs better in binary classification. So to test model using SVM, it was trained with three different kernel functions. SVM was trained and tested with same dataset as that of FFNN. Table 5 shows confusion matrix for three different kernel functions of SVM. It is evident from the results that RBF performs better than other two kernel function in terms of overall misclassification. Therefore SVM with RBF kernel is used for further performance evaluation. Table 6 lists the accuracy of the three kernel functions. Clearly, SVM if trained with RBF kernel has least false positive (FP) rate. Fig. 8 shows distribution of difference between fault and true value of

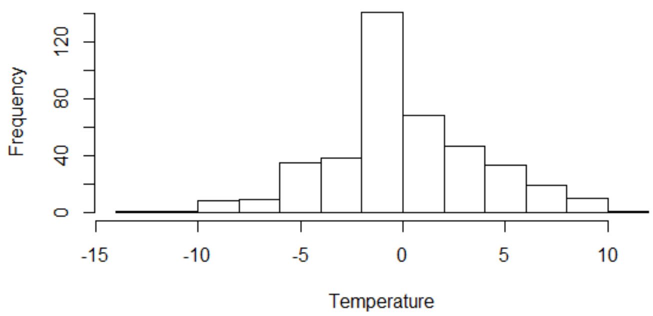


Figure 7: Histogram of distance of misclassified points from true data using FFNN

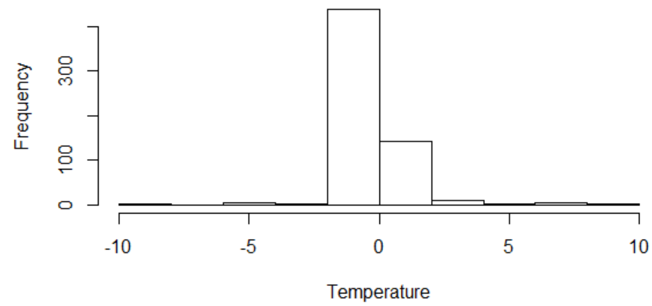


Figure 8: Histogram of distance of misclassified points from true data using SVM with RBF kernel

misclassified points by SVM using RBF kernel. Exactly 606 points were misclassified and out of which 562(92.7% of the misclassified points) lie away from true data by less than 0.5°C. So only 44(7.2% of the misclassified points) lie away from true data more than 0.5°C. Total misclassified (44) points is just 0.2% of the total test data. This suggests that out of 1000 measurements model will be wrong about only 2 times which is better than FFNN. Possibility for large error before applying slack of 0.5°C in SVM could be due to margin around SVM learnt classification plane is smaller than 0.5°C.

Table 5: Confusion matrix with time window size 5 of FR sensor on test dataset using SVM

Kernel	Actual label	Classified TRUE	Classified FALSE
Linear	TRUE	19517	2
	FALSE	1600	2192
RBF	TRUE	18967	552
	FALSE	54	3738
Cubic polynomial	TRUE	19519	0
	FALSE	2424	1368

Table 6: Accuracy of SVM with three different types of kernel

SVM kernel	TP rate	FP rate	F1 score
Linear	1.00	0.42	0.96
RBF	0.97	0.01	0.98
Degree 3 poly	1.00	0.64	0.76

4.4 Generalization of model

FFNN and SVM were also tested with data of two different sensor nodes. The clean data from these sensors was injected with some errors defined earlier and this data was used for testing with models. Two different sensors are namely FL and BL. FL sensor is highly correlated with FR but BL is poorly correlated with FR.

Table 7 shows confusion matrix of testing with FL data. It shows that only 109(0.6% of the total dataset(18261)) points are misclassified. In this case, error rate is better than FR test data itself. Moreover, BL dataset was tested and table 8 shows confusion matrix of the same. Accuracy is quite lower than FR in this case, as BL exhibits some patterns that are different than FR.

SVM was also tested with data of two different sensor nodes. Table 7 shows confusion matrix of FL data, even though FL is highly correlated with FR sensor SVM has very low accuracy on this dataset. Exactly 2692(15% of the total dataset) points were misclassified by SVM for FL data. Table 8 shows confusion matrix for BL data and performance almost same as FFNN on BL data. Exactly 1326(6.3% of the total dataset) points were misclassified by SVM for BL data.

Table 7: Confusion matrix with time window size 5 of FL sensor data using FFNN and SVM

Method	Actual label	Classified TRUE	Classified FALSE
FFNN	TRUE	17466	7
	FALSE	102	686
SVM RBF	TRUE	14883	2590
	FALSE	102	686

Table 8: Confusion matrix with time window size 5 of BL sensor data using FFNN and SVM

Method	Label label	Classified TRUE	Classified FALSE
FFNN	Actual TRUE	17761	918
	Actual FALSE	328	2198
SVM RBF	Actual TRUE	17427	1252
	Actual FALSE	74	2452

4.5 Comparison of methods

Previous sections discuss the results obtained with two different modeling techniques namely FFNN and SVM. Table 9 shows comparison between two methods on the same training and test dataset.

SVM has better error rate compared with FFNN for points that are not away more than 0.5°C from true data but otherwise FFNN has low error rate than SVM. Also training time of the SVM is about 30 times more than FFNN. As evident from figures 7 and 8 after considering slack of 0.5°C SVM has smaller variance in the error than FFNN. FFNN is advantageous over SVM only if computationally simpler model is required as SVM with RBF kernel function is little heavier than FFNN for computing results. But for the purpose of current work neural network is well suitable as computation time is not very significant considering sampling time of every sensor being 5 minutes but for SVM training time and computation time is large and also it does not generalize well to other sensors. Also considering performance on other sensor's data, FFNN has performed better on both FL and BL data, where as SVM has quite lower accuracy on correlated sensor which is a surprise and needs more analysis to find out exact reason.

Table 9: Comparison between performance of FFNN and SVM on the same train and test dataset

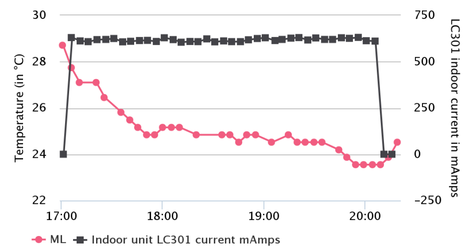
Parameters	FFNN	SVM (RBF Kernel)
Error rate	1.8%	6.7%
Error rate with difference ≥ 0.5°C	0.7%	0.2%
Time to train model	98 sec	3000 sec

5 SMARTER CONSUMPTION AND SOFTWARE PLATFORM

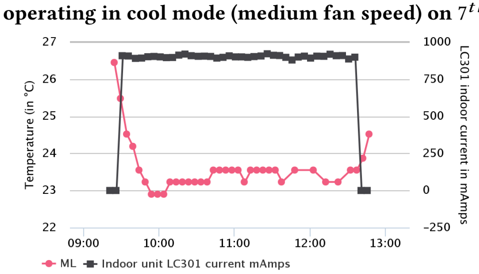
The data gathered from the WSN and the energy meters is a rich source of information regarding energy consumption patterns and its impact on the environment of LC301 classroom and vice versa. A detailed analysis on this data has provided valuable insights on how to detect instances of energy wastage as well as how to reduce end-use consumption without impacting the comfort of the occupants. We present specific details in the following subsections.

5.1 Detection of Energy Wastage and Optimization

We expect lights and ACs inside the classroom to be ON only during scheduled class timings. Hence, a simple check of lights being OFF and ACs being ON is applied to detect AC energy wastage. For some instances, lights were switched OFF at around 20:20 Hrs but the ACs were still operating till 1:00 Hrs. This is a clear indication of 5hrs of AC energy wastage. As per this analysis, $\approx 82kWh$ of AC energy wastage was detected from 20th August to 20th October 2015. which is equivalent to AC energy consumption of ≈ 8 lectures of 1hr each. PIR sensors can be installed on every mote to check the occupancy in case both lights and ACs are ON. This comes at the cost of battery life and increased overall system cost. Another solution that we are exploring is to determine the dependency of the rate of change in the outdoor unitfis consumption on the occupancy inside the classroom. The AC can be operated in cool and dry mode with adjustable fan speeds. Each mode is characterized by different power consumption and temperature gradients. Our investigations indicate that internal temperature quickly converges to set temperature in cool mode with high fan speed, which results in high current consumption for indoor units as well as high power consumption by outdoor unit. We observed that the indoor unit consumption is higher for higher fan speeds. A corresponding trend also persists in the energy consumed by outdoor unit (5.7 kWh per hour for medium fan speed versus 10 kWh per hour for high fan speed). Correspondingly, the temperature gradient is lower as fan speed decreases. We claim that the environment monitoring should be suitably integrated into an energy optimization module wherein control signals can be sent to the AC units to switch between modes with different fan speed to attain comfortable temperatures while minimizing energy consumption. For instance, if the internal temperature before the AC is switched on is high, then the ACs can be run in cool mode with high fan speed to achieve set temperature quickly after which the temperature can be maintained in a more economical mode such as cool mode with low fan speed. The following section further expands on these ideas.



(a) AC operating in cool mode (medium fan speed) on 7^{th}Sept 15



(b) AC operating in cool mode (high fan speed) on 14^{th}Sept 15

Figure 9: Indoor unit operation in different modes and temperature variations

5.2 Data Visualization and Analysis

Multiple studies have expounded on the benefits of visualization in decision making. With this in mind, we have developed a user interface wherein measurement data stored in the server can be suitably displayed as per the user requests. The online interface is very interactive and helps in visualizing the data in real time.

The web interface has a secure web-page for LC301 classroom showing plots of temperature, humidity, luminosity and battery consumption in real time. The interface also shows current consumption of indoor AC units, power consumption of outdoor AC unit and lights/fans in real-time. A snapshot of the interface is shown in Fig. 10. Legends at the bottom of the chart can be selected to enable or disable the corresponding plot.

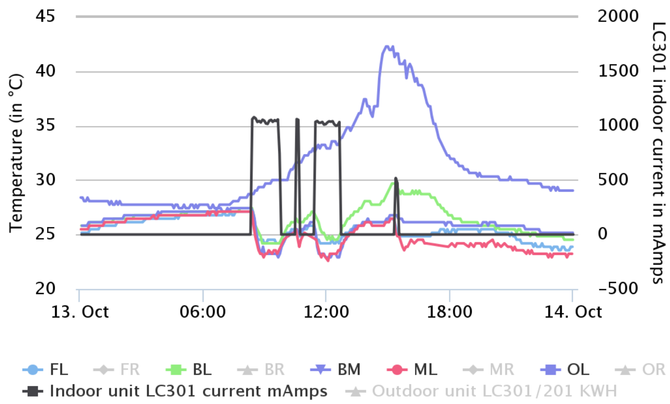
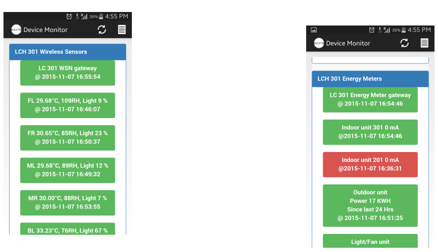


Figure 10: Temperature data of sensors with selectable legends

If sensor motes, energy meters or even gateway fail to communicate data due to connectivity problems, the users are automatically notified on the web interface. Failure alerts are also communicated to a corresponding Android App. Snapshots of App interface during normal operations and during an instance of device failure is provided in Fig.11. The web interface and Android App clearly notify the failed sensor, its location as well as the cause of failure. This makes tracking and debugging devices in real-time possible. As this network expands throughout the campus, such functionality will help in identifying device failure quickly.



(a) Under normal conditions

(b) When a meter fails

Figure 11: Android App interface wherein red background indicates device failure.

The web interface has a dedicated web-page for the analysis of collected data. The platform allows users to: (a) calculate statistical parameters on a series of data, (b) generate a scatter plot of two different measurements and (c) plot data of different sensors in time-series. Such analysis is very useful and can be performed in real-time or on historical data stored in server. The interface also provides an option to save the analytical work in the form of images (PNG, JPG, SVG format) or PDF format. Interface has REST API developed in PHP language. Any programming language script can make secure HTTP request to REST API for the data with appropriate credentials and interface provides data in JSON format upon validating credentials. This functionality helps user in accessing data from any where on the Internet.

5.3 Energy Management

To avoid energy wastage, alerts are generated when the AC is ON with very low luminosity inside the classroom or AC is unnecessarily running in high power mode. Operators in lecture hall complex are alerted via the Android App as well as the web-interface in real-time to take necessary action. For instance, if the temperature and humidity measurements indicate that a sufficiently comfortable environment has been attained, a suitable alert is generated and flashed on the web interface as well as the Android app. Such alerts may be directly translated into appropriate control signals for the AC system.

A missing link in the current interface is an energy demand model, which captures the variation of energy consumption with environment and occupancy. Our current research is exploring tree- and neural network-based models for this purpose [3]. The goal is to apply the model for energy prediction and suitably integrate it into a comprehensive energy optimization module.

6 CONCLUSION

This paper presents a smart analytics platform based on real-time WSN data which could be used to optimize the performance and energy efficiency of buildings. The platform has an interactive interface displaying real-time energy consumption and can notify alerts as needed. A salient feature of this platform is its ability to flag faulty sensor measurements with a very high accuracy. We hope that such an online software platform can enable consumers with decision support capabilities for better energy management and sensitize then to consume resources more efficiently.

Future work will explore modeling of AC energy consumption for purposes of prediction and control. We will also look at options for closing the feedback loop with AC system so that alerts can directly translate into energy savings. Finally, we will explore the potential of this enabling technology for extracting demand response from end-users.

REFERENCES

[1] Edward Arens, CC Federspiel, Danni Wang, and Charlie Huizenga. 2005. How ambient intelligence will improve habitability and energy efficiency in buildings. In *Ambient intelligence*. Springer, 63–80.

[2] Sandra Sendra Compte, Jaime Lloret, Miguel García Pineda, and José Francisco Toledo Alarcón. 2011. Power saving and energy optimization techniques for Wireless Sensor Networks. In *Journal of communications*, Vol. 6. Engineering and Technology Publishing, 439–459.

[3] Anastasios I Dounis and Christos Caraiscos. 2009. Advanced control systems engineering for energy and comfort management in a building environmentfi!?A review. *Renewable and Sustainable Energy Reviews* 13, 6 (2009), 1246–1261.

[4] Ahmad Faruqui, Sanem Sergici, and Ahmed Sharif. 2010. The impact of informational feedback on energy consumptionfi!?A survey of the experimental evidence. *Energy* 35, 4 (2010), 1598–1608.

[5] Paul M Frank. 1990. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *automatica* 26, 3 (1990), 459–474.

[6] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin.

[7] DJ Hammerstrom, R Ambrosio, J Brous, TA Carlon, DP Chassin, JG DeSteese, RT Guttromson, GR Horst, OM Järvegren, R Kajfasz, and others. 2007. Pacific northwest gridwise testbed demonstration projects. *Part I. Olympic Peninsula Project* 210 (2007).

[8] Georg Jager, Sebastian Zug, Tino Brade, Andre Dietrich, Christoph Steup, Christian Moewes, and Ana-Maria Cretu. 2014. Assessing neural networks for sensor fault detection. In *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2014 IEEE International Conference on*. IEEE, 70–75.

[9] Michael Kintner-Meyer. 2005. Opportunities of wireless sensors and controls for building operation. *Energy engineering* 102, 5 (2005), 27–48.

[10] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli. 2003. On-line fault detection of sensor measurements. In *Proceedings of IEEE Sensors 2003 (IEEE Cat. No.03CH37498)*, Vol. 2. 974–979 Vol.2. DOI: http://dx.doi.org/10.1109/ICSENS.2003.1279088

[11] Tuan Anh Nguyen and Marco Aiello. 2013. Energy intelligent buildings based on user activity: A survey. *Energy and buildings* 56 (2013), 244–257.

[12] Kevin Ni, Nithya Ramanathan, Mohamed Nabil Hajj Chehade, Laura Balzano, Sheela Nair, Sadaf Zahedi, Eddie Kohler, Greg Pottie, Mark Hansen, and Mani Srivastava. 2009. Sensor Network Data Fault Types. *ACM Trans. Sen. Netw.* 5, 3, Article 25 (June 2009), 29 pages. DOI: http://dx.doi.org/10.1145/1525856.1525863

[13] Lilia Paradis and Qi Han. 2007. A Survey of Fault Management in Wireless Sensor Networks. *Journal of Network and Systems Management* 15, 2 (2007), 171–190. DOI: http://dx.doi.org/10.1007/s10922-007-9062-0

[14] Xiao Xu, J Wesley Hines, and Robert E Uhrig. 1999. Sensor validation and fault detection using neural networks. In *Proc. Maintenance and Reliability Conference (MARCON 99)*. 10–12.

[15] Yi Yu and Saleem N Bhatti. 2014. The cost of virtue: reward as well as feedback are required to reduce user ICT power consumption. In *Proceedings of the 5th international conference on Future energy systems*. ACM, 157–169.

[16] Y. Zhang, N. Meratnia, and P. Havinga. 2010. Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys Tutorials* 12, 2 (Second 2010), 159–170. DOI: http://dx.doi.org/10.1109/SURV.2010.021510.00088

[17] Sebastian Zug, André Dietrich, and Jörg Kaiser. 2012. Fault-handling in networked sensor systems. *Fault Diagnosis in Robotic and Industrial Systems* (2012).