PNNL-25996

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

Development and Evaluation of Algorithms to Improve Small- and Medium-Size Commercial Building Operations

October 2016

W Kim
S Katipamula
RG Lutes
RM Underhill

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov


This document was printed on recycled paper.
(8/2010)

PNNL-25996

Development and Evaluation of Algorithms to Improve Small- and Medium-Size Commercial Building Operations

WH Kim
S Katipamula
RG Lutes
R Underhill

October 2016

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

Small- and medium-sized (<100,000 sf) commercial buildings (SMBs) represent over 95% of the U.S. commercial building stock and consume over 60% of total site energy. Many of these buildings use rudimentary controls that are mostly manual, with limited scheduling capability and no monitoring or failure management. Therefore, many of them are operated inefficiently and consume excess energy. SMBs typically use packaged rooftop units (RTUs) that are controlled by an individual thermostat.

Many reasons drive the increased urgency to improve the operating efficiency of the existing U.S. commercial building stock; chief among them is the need to mitigate climate change impacts. Studies have shown that managing set points and schedules of the RTUs will result in up to 20% energy and cost savings. Another problem associated with RTUs is short cycling, where an RTU goes through ON and OFF cycles too frequently. Excessive cycling can lead to excessive wear and hence premature failure of the compressor or its components. Short cycling can result in a significantly decreased average efficiency (up to 10%), even if there are no physical failures in the equipment. Also, SMBs use a time-of-day scheduling is to start the RTUs before the building is occupied and shut it off when it is unoccupied. Ensuring correct use of the zone set points and eliminating frequent cycling of RTUs, thereby leading to persistent building operations, can significantly increase the operational efficiency of SMBs. A growing trend is to use low-cost control infrastructure that can enable scalable and cost-effective intelligent building operations.

The work documented in this report describes three algorithms for detecting the zone set point temperature, RTU cycling rate, and occupancy schedule that can be deployed on low-cost infrastructure. These algorithms only require the zone temperature data for detection. The algorithms have been tested and validated using field data from a number of RTUs from six buildings in different climate locations. Overall, the algorithms were successful in detecting the set points and ON/OFF cycles accurately using the peak detection technique, and detecting the occupancy schedule using a symbolic aggregate approximation technique.

This report describes the three algorithms, results from testing the algorithms using field data, how the algorithms can be used to improve SMBs efficiency, and presents related conclusions.

Acknowledgments

The authors acknowledge the Buildings Technologies Office of the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy for supporting this research and development effort. In particular, the authors thank Dr. Marina Sofos, Technology Development Manager, for her guidance and strong support of this work. At PNNL, we acknowledge George Hernandez for his technical guidance, Andrew Nicholls for his thoughtful comments and insights, and Susan Ennor for editorial support in preparing this document.

Acronyms and Abbreviations

BAS	building automated system
HVAC	heating, ventilation, and air-conditioning
Hz	hertz
MAPE	mean absolute percent error
PAA	Piecewise Aggregate Approximation
PNNL	Pacific Northwest National Laboratory
RTU	rooftop unit (air-conditioner)
SAX	Symbolic Aggregate Approximation
SMB	small- and medium-sized building

Nomenclature

$Accuracy\ matrics$	accuracy of confusion matrix	
$Actual$	a number of actual on/off cycling	[ea]
BAS	building Automation System	
C_0	occupied schedule	
C_1	unoccupied schedule	
$HVAC$	heating, ventilation, and air-conditioning	
$Identified$	a number of on/off cycling perdition	[ea]
$MINDIST$	minimum distance between two string formats	
$N_{0,0}$	number of correctly identified C_0	[ea]
$N_{1,0}$	number of C_1 incorrectly identified as C_0	[ea]
$N_{0,1}$	number of C_0 incorrectly identified as C_1	[ea]
$N_{1,1}$	number of correctly identified C_1	[ea]
n	a number of predictions	[ea]
$P(x)$	mixture density function	
$p(x\|\omega_i)$	conditional probability of x	
$P(w_i\|x)$	conditional probability of w_i	
$P(w_i)$	prior density function	
T_i	PAA representation	[˚F]
T_n	normalized temperature	[˚F]
T_{max}	local maximum temperature	[˚F]
T_{min}	local minimum temperature	[˚F]
T_{peak}	array of maximum temperature	[˚F]
T_{oa}	outdoor air temperature	[˚F]
T_{sp}	set point temperature	[˚F]
T_{zone}	array (collection of data) of temperature measurements for a zone served by an RTU	[˚F]
T_{valley}	array of minimum temperature	[˚F]
s	standard deviation of prediction	
t	time series	
t_c	t-value for corresponding confidence level	
x_o	threshold partitioned into two regions, R1 and R2	
$\bar{x}$	mean of prediction	[˚F]
w	segment number	[ea]

Contents

Figures

Tables

1.0 Introduction

Small- and medium-sized (<100,000 sf) commercial buildings (SMBs) represent over 95% of the U.S. commercial building stock and consume over 60% of total site energy (EIA 2012). These buildings generally do not have a dedicated building operator or an energy manager and therefore, the heating, ventilation, and air-conditioning (HVAC) equipment tends to be serviced as a result of occupant complaint or when the units fail. Even buildings that get periodic maintenance have a number of operational problems that go undetected. The operational problems result from improper control or incorrect commissioning, which leads to inefficient operation, increased energy use, and reduced equipment life (Mařík et al. 2011; Mills 2011; Wang et al. 2013). Several studies have documented that commercial buildings consume between 10% and 30% excess energy because of operational problems or performance degradation (Ardehali et al. 2002; Katipamula and Brambley 2004a,b; Breuker and Braun 1998; and Jacobs 2003a). Therefore, improvements in HVAC system operations can lead to significant reductions in energy use and greenhouse gas emissions.

Over 85% of the commercial buildings lack building automation systems (BASs), and most of these buildings are SMBs; they represent about 57% of the total commercial building area and consume about 46% of the total site energy (EIA 2012). These buildings employ rudimentary controls that are mostly manual, with limited scheduling capability and no monitoring or failure management. Therefore, many of these buildings are operated inefficiently and consume excess energy. There are a number of reasons why these buildings do not deploy BASs: 1) lack of awareness, 2) lack of inexpensive packaged solutions, and 3) sometimes the owner is not the tenant and has no incentive to invest in a BAS (Katipamula et al. 2012). SMBs without BASs may have limited ability to monitor or trend the data necessary for detecting system degradation or for performing supervisory controls. SMBs typically use packaged rooftop units (RTUs) that are controlled by an individual thermostat. The RTUs are often maintained poorly and degradation of performance and faults are only addressed when occupants complain or a unit fails. Cowan (2004) conducted a survey and analysis of 503 RTUs and found that 54% of the RTUs had problems: 42% exhibited improper airflow, 72% improper refrigerant charge, and 20% failed sensors. These problems led to an estimated excess energy consumption of 8%. Another study evaluated 109 RTUs in the field and found that 89 had fault conditions—31 had two or more faults (ADM 2009). The average energy efficiency ratio for the units increased from 6.6 before servicing to 7.0 after servicing, an average increase of over 6%.

Another problem associated with RTUs is short cycling, an operational mode during which an RTU goes through ON and OFF cycles too frequently. Short cycling can be caused by equipment oversizing, poor thermostat location, low refrigerant charge, a clogged air filter, and other reasons. Excessive cycling of RTUs can lead to excessive wear and premature failure of the compressor or its components. Also, short cycling can result in a significantly decreased average efficiency, even if there are no physical failures in the equipment. According to the Small HVAC System Design Guide (Jacobs 2003b), the short cycling of RTUs can cause an efficiency penalty of approximately 10%. Hence, detecting the zone set point and frequent cycling is the first step in analyzing system performance, improving efficiency, and ensuring the desired persistence of building operations. There is a growing trend to use low-cost control infrastructure that can enable scalable and cost-effective intelligent building operations (Katipamula et al. 2016a,b). The inexpensive controls infrastructure can be leveraged to provide opportunities for control coordination and embedded diagnostics for SMBs.

The primary objectives of the project conducted by Pacific Northwest National Laboratory (PNNL) and reported here were to:

- Design and develop three algorithms that can detect 1) zone set points, 2) frequent compressor ON/OFF cycling, and 3) the occupancy schedule based only on zone temperature measurement.
- Program the algorithms in Python, enabling configuration and execution of these algorithms on the VOLTTRON™ platform.
- Validate the performance of the three algorithms based on data from 26 RTUs from six different locations.

The ensuing sections of this report describe the three algorithms for detecting the zone set point temperature, RTU cycling, and occupancy schedules that are compatible for deployment on low-cost controls infrastructure. These algorithms only require the zone temperature data for detection purposes. In addition, the output of the algorithms can be used as the control variable for analyzing system performance and developing more robust control methodologies that can be incorporated into existing building systems. The algorithms have been widely tested and validated using data from a number of RTUs in the field. The remainder of the report describes the results from testing the algorithms and presents related conclusions.

2.0 Description of Algorithms

This section provides a brief description of the three algorithms for zone set point detection, ON/OFF cycling detection, and occupancy schedule detection. Detecting the zone set point, frequent cycling, and occupancy schedules is the first step in analyzing system performance, improving efficiency, and ensuring persistence of building operations. There is a growing trend to use low-cost control infrastructure that can enable scalable and cost-effective intelligent building operations (Katipamula et al. 2016a,b). The inexpensive controls infrastructure can be leveraged to provide opportunities for control coordination and embedded diagnostics for SMBs.

2.1 Zone Set Point Temperature Detection

Some SMBs without BASs use programmable thermostats with a predefined control sequence that is executed based on some inputs (e.g., sensor data, time-of-day). The thermostat contains a temperature sensor (typically a thermistor) that measures the zone temperature and compares that temperature to a set point. In cooling mode, when the temperature exceeds the set point (plus dead band) the thermostat will send a signal to the RTU to activate cooling. Conversely, when the temperature drops below the set point (minus dead band) the thermostat will send a signal to the RTU to activate heating. A programmable thermostat can vary the temperature set point based on time-of-day and day-of-week, allowing for night setup or setback. However, the thermostat has limited ability to monitor or trend the data necessary for detecting system degradation or for use in control optimization. For instance, the thermostat often uses a single set point for each operating mode (e.g., occupied, unoccupied, heating, cooling, etc.). Many SMBs either do not have programmable thermostats or the thermostats are not programmed accurately to reflect the desired schedule or set point. Enforcing set points and schedules can result in energy savings of over 20% (Katipamula et al. 2012).

2.2 ON/OFF Cycling Detection

Another problem associated with RTUs is short cycling, an operational mode during which an RTU goes through ON and OFF cycles too frequently. Short cycling can be caused by equipment oversizing, poor thermostat location, low refrigerant charge, clogged air filter, and other reasons. Excessive cycling of RTUs can lead to excessive wear and premature failure of the compressor or its components. Short cycling can also result in a significantly decreased average efficiency, even if there are no physical failures in the equipment. According to the Small HVAC System Design Guide (Jacobs 2003b), the short cycling of RTUs can cause an efficiency penalty of approximately 10%. Peak and valley detection can be used to quantify the number of times an RTU is ON cycling or OFF cycling in a given time period.

2.3 Occupancy Schedule Detection

Building occupants are critical determinants of energy consumption. Proper occupancy scheduling is one of the most simple control strategies for reducing energy costs in existing buildings (Murphy and Maldeis 2009). Some SMBs without BASs use a programmable thermostat with time-of-day scheduling. The most common uses of time-of-day scheduling is to start the HVAC system before the building is occupied and to shut it off after the building is unoccupied.

The programmable thermostats can vary the set point temperature based on time-of-day and day-of-the-week. For example, when the building is scheduled to be occupied, the programmable thermostat controls the HVAC system to maintain the zone temperature within a desired comfort level. When the building is

scheduled to be unoccupied, the programmable thermostat resets the zone temperature to a value that is different from the occupied set point by a fixed amount, called night setback or setup. However, the programmable thermostat allows the occupants to override the schedule or change set points by using the temperature "hold" feature of the thermostat.

It is challenging to recognize these overrides in set points and schedule changes without a BAS because thermostats are seldom networked. These temporary changes often remain permanent and can result in unnecessary excess energy use. Therefore, a simple method of detecting the occupancy schedule is required to automatically ensure the HVAC system is operating as intended and to identify improper schedule changes to maintain the energy savings expected by using proper time-of-day scheduling. Other opportunities also exist for using the occupancy scheduling detection approach to save energy in buildings. The occupancy schedule information can be inputs to other local control systems to optimize the control of the HVAC system. For example, when time-of-day scheduling is used, the HVAC system should start prior to occupancy and operate long enough for the zone temperature to reach the desired zone temperature. The optimal start approach is used to determine the length of time required to reach the occupancy set point temperature prior to occupancy time. The beginning of the occupied schedule can be used as input for the optimal start approach.

Previous studies used a stochastic model to learn occupancy patterns and predict the occupancy schedule. Tebak and Vries (2010) proposed an occupancy schedule model using a probabilistic formula to predict the occurrence and frequency of intermediate break activities during an occupied time. Sun et al. (2014) developed a stochastic model based on an exponential distribution to predict the duration of overtime periods during unoccupied time. Stoppel and Leite (2014) presented a probabilistic-based model to predict the occupancy schedule for hourly occupant presence for weekday and weekend time periods. In recent years, data mining has gained popularity in the building science area, including mining data about occupant behavior, fault detection, and building energy performance. D' Oca et al. (2015) used data mining approaches to discover individualized occupancy schedules in office buildings. A data set of 16 offices with 10-minute interval occupancy data over a 2-year period is mined through a decision tree model that predicts the occupancy presence. The cluster analysis is employed to obtain consistent patterns of occupancy schedules representative of typical single office working user profiles. However, previous studies based on data mining mostly required large measurement data sets to identify the occupancy schedule in the building. The method developed by PNNL and reported herein uses Symbolic Aggregate Approximation (SAX) and clustering to detect the occupancy structure of zone temperature data. The SAX method was developed by Lin et al. (2003) to mine information from time-series data. It is widely employed in pattern identification, sequence classification, abnormality detection, and other data mining research (Lin et al. 2007).

3.0 Field Test Data Used for Validation of the Algorithms

To test and validate the PNNL-developed algorithms, field test data from 24 RTUs (air conditioners and heat pumps) from six different locations in the United States were used, as summarized in Table 3.1. Building A and Building B are office buildings located in suburban of Seattle, Washington, and Berkeley, California. Building C is a mechanical shop building with offices located in Richland, Washington. Building D is a machine product factory located in a suburb of South Paris, Maine. Building E and Building F are fitness centers located in Miami, Florida, and Cupertino, California. Buildings in different climate zones and local conditions were selected to evaluate the performance of the algorithms.

The different data sets from each RTU were selected from existing field data based on the outdoor air temperature conditions (i.e., spring/fall, summer, and winter). Each data set contains five measurements (zone temperature, set point temperature, supply fan status, cooling/heating command, and outdoor temperature) from a 1-week period. Although the algorithms used only zone temperature, the remaining sensor data are used as "ground truth" data to verify the results of the algorithm.

Table 3.1. Validation Data Set for Six Different Locations

Building	Location	Number of RTUs	Summer Data Set $T_{OA,avg}$ (°F)	Winter Data Set $T_{OA,avg}$ (°F)	Spring/Fall Data Set $T_{OA,avg}$ (°F)
A	Seattle, WA	4	77	42	54
B	Berkeley, CA	6	75	53	64
C	Richland, WA	8	76	47	66
D	South Paris, ME	2	73	-	61
E	Miami, FL	2	84	69	79
F	Cupertino, CA	2	73	50	60

4.0 Development and Validation of Zone Set Point Temperature Detection Algorithm

This section describes the development of the set point detection algorithm using only zone temperature measurement as an input. The algorithm consists of three parts: preprocessing, peak detection, and classification. During preprocessing, the noise associated with the raw temperature data are filtered using a low pass filter. Next, peak detection is used to detect peaks in the data and to measure their positions, heights, and widths. The output of the peak detection is processed by a Bayesian classifier, which compares two adjacent set points to confirm whether they are same or different. The set point algorithm has been coded in Python and integrated with VOLTTRON, and is publicly available as part of the VOLTTRON distribution.

4.1 Low Pass Filter for Preprocessor

The presence of random noise in measured zone temperature data will cause false set point detection simply due to the noise. A low pass filter is employed during preprocessing (Orfanidis 1995; Smith 2013) to remove the noise. In signal processing, the low pass filter is a process that removes frequencies that are higher than a certain preset frequency threshold. The low pass filter can eliminate the noise in the zone temperature measurement that would potentially be incorrectly classified as the set point. The temperature measurement that is acquired in the time domain is converted to a frequency domain signal in which the independent variable is frequency rather than time.

The Z-transform is performed on the time signals. It converts a discrete time domain signal into a complex frequency domain representation. The Z-transformation is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} X(n) Z^{-n} \tag{1}$$

where z is the complex number and X(n) is the time-series signal.

The low pass filter allows low-frequency temperature from 0 Hz to the cutoff frequency while blocking frequencies higher than the cutoff frequency. Selection of a large cutoff frequency can lead to more stable states, but less input data for set point detection. On the other hand, smaller cutoff frequency increases the uncertainty of the set point detection outputs. Therefore, it is necessary to find cutoffs that minimize the uncertainty of the set point detection while maximizing the use of input data. The appropriate cutoff frequency is chosen based on the review of the time-series temperature data from several buildings. The frequency domain signal is then multiplied by the frequency response of a digital low pass filter that removes frequency components. The inverse Fourier transform then recovers the filtered time domain spectrum.

4.2 Peak Detection Algorithm

The detection of peaks in signals is an important step in many signal processing applications. The peak detection algorithm is used in nuclear monitoring (Azzini et al. 2004), mass spectrometry (Coombes et al. 2005), processing (Jordanov et al. 2002; Harmer et al. 2008), and electronic systems (Schneider 2011). Peak (Valley) points denote the significant events where the function graph changes from increasing

(decreasing) behavior to decreasing (increasing) behavior in a time series (Palshikar 2009). The identification of these behaviors is important for analysis of the time-series data.

Figure 4.1 shows the example of preprocessed zone temperature of an RTU during the summer season. When the zone temperature exceeds the cooling temperature set point (plus dead band), the zone temperature will reach a local maximum (denoted as T_{max}, represented by the blue circles in Figure 4.1). As the RTU turns ON and provides cooling to the zone, the temperature will decrease until the zone temperature falls below the cooling temperature set point (minus dead band), at which time the RTU will stop cooling (local minimum denoted as T_{min}, represented by the red circles in Figure 4.1). The heat pump operation in the heating mode is similar to the operation of a cooling mode. The difference is the sequence of local maximum and minimum. For example, the zone temperature will reach a heating temperature set point minus a dead band (local minimum denoted as T_{min}) and then as the heat pump system provides heating to the zone the temperature will rise until the zone temperature is above a heating temperature set point plus dead band (denoted as T_{max}), at which time the heat pump system will stop the heating operation.

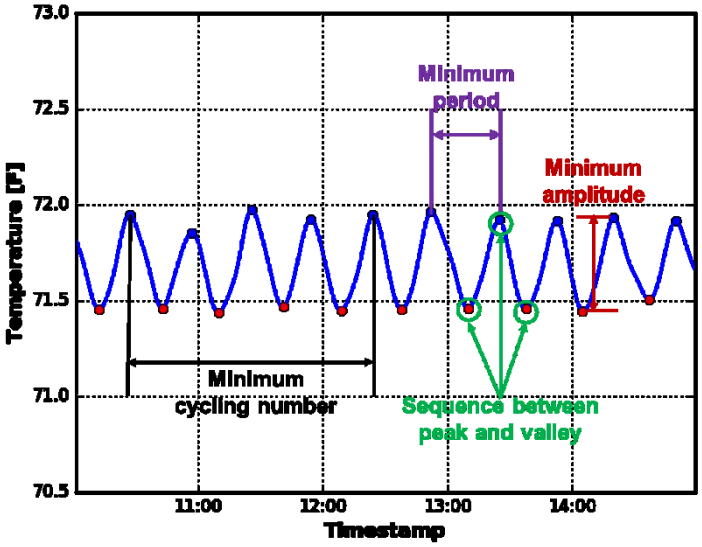

Figure 4.1. Zone Temperature and Four Criteria Used for Set Point Detection

Peak detection uses the approach of finding the locations and amplitudes of T_{min} and T_{max}. Let T_{zone} be a given array of zone temperatures that represents the time-series data (Cormen 2009). A way to detect T_{max} in T_{zone} is to use the property that a peak must be greater than its immediate neighbors. For example, given array T_{zone} with n elements, the algorithm finds the index i of peak element $T_{zone}[i]$ where $T_{zone}[i] \geq T_{zone}[i-1]$ and $T_{zone}[i] \geq T_{zone}[i+1]$ as shown in Equation (2).

$$T_{peak} = T_{zone}[i-1] < T_{zone}[i] > T_{zone}[i+1], \quad i = 2, 3, \cdots, n-1 \tag{2}$$

For elements (i=1 or i=n) on the boundaries of the array, the element only needs to be greater than or equal to its lone neighbor to be considered a peak. The array of maximum temperatures (T_{peak}) is defined as shown in Equation (3).

$$\begin{aligned} T_{peak} &= T_{zone}[1] > T_{zone}[2] \\ T_{peak} &= T_{zone}[n-1] < T_{zone}[n] \end{aligned} \tag{3}$$

In contrast, we define the array of minimum temperatures (T_{valley}) as shown in Equation (4).

$$T_{valley} = T_{zone}[i-1] > T_{zone}[i] < T_{zone}[i+1], \;\; i = 2, 3, \cdots, n-1$$
$$T_{valley} = T_{zone}[1] < T_{zone}[2] \tag{4}$$
$$T_{valley} = T_{zone}[n-1] > T_{zone}[n]$$

A key issue for the peak detection algorithm is the fact that peaks occur with different amplitudes and at different scales, which results in a large number of false positives among detected peaks. To filter out valid peak information or fail to reject false peaks, the algorithm can measure the position, height, width, and sequence of each peak. These measurements are compared to preset criteria to reduce identification of false peaks and ensure that the locations and amplitude results are reasonable while finding peaks and valleys. For example, it is possible to detect only the desired peaks and ignore peaks that are too small, too wide, or too narrow. A description of each threshold follows below.

4.2.1 Minimum Period between Neighboring Peaks

All RTU compressors need a minimum of 5 minutes of continuous run time to saturate bearings, warm surfaces, and equalize refrigerant side pressure. The compressor ON/OFF signals are valid if the time difference between the ON signal and OFF signal is at least 5 minutes. Therefore, the minimum period is set as 5 minutes as the default value (purple line shown in Figure 4.1). The period between the T_{max} (T_{min}) and neighboring T_{min} (T_{max}) must be greater than the minimum period. The minimum period may have to be adjusted based on the thermostat's time delay or equipment internal setting.

4.2.2 Sequence between Peak and Valley

Compressor equipment ON-OFF cycling should be a sequential order of events (e.g., OFF signal → ON signal→ OFF signal). For instance, a valid T_{max} point will be adjacent to two T_{min} points as shown in Figure 4.1. The same signals (two adjacent T_{max}) that are repeated can be removed from T_{peak} (T_{valley}). For example, the repeated ON-signal detection is considered to represent continuous compressor running (e.g., ON signal → ON signal→ ON signal → ON signals).

4.2.3 Minimum Amplitude between Peak and Valley

T_{max} (T_{min}) can be ignored if the amplitude between T_{max} (T_{min}) and neighboring T_{min} (T_{max}) is below minimum amplitude (red line as shown in Figure 4.1). When the amplitude is kept very small, all of the peaks including T_{max} that resulted from noise in the temperature can be detected. When the difference is kept very large, it is difficult to detect T_{max}. The suggested default for the minimum difference is 0.3°F. For example, T_{max} is valid if the temperature difference between T_{max} and neighboring T_{min} is higher than 0.3°F. The minimum amplitude can be adjusted based on the thermostat's dead band value.

4.2.4 Minimum Cycling Number

The minimum cycling number is a minimum number of T_{max} and T_{min} per day required for accurate set point detection. If the number of T_{max} and T_{min} is less than this value, any T_{max} and T_{min} are ignored. The minimum number of cycling of 5/day is the suggested default as shown in Figure 4.1. For example, if the minimum cycling number is smaller than 5/day, then the set point detection indicates that the minimum number of cycles is not enough to detect the set point temperature.

Table 4.2 shows the example of T_{max}, T_{min}, and temperature set point (denoted $T_{sp,pred}$). The set point detection algorithm uses a fixed-length moving average for detecting set point temperature. The moving average is obtained by taking the averages of fixed subsets of the number series. In this approach, the

number of fixed subsets (n) is set to be 5. For example, each $T_{sp,pred}$ can be determined by averaging the zone temperature reading at the T_{max} and T_{min}. Five $T_{sp,pred}$ in the fixed subsets are replaced by their average value ($T_{sp,avg}$) of the data points calculated using Equation (5). Then, the subset is modified by excluding the first number in the series and including the next number following the original subset in the series. This process is repeated over the entire data series. In this example, $T_{sp,avg}$ for the fixed window is 71.7°F, as shown in Table 4.2.

$$T_{sp,avg} = \frac{\sum_{i=1}^{n}\left(\frac{\left(T_{max,i} + T_{min,i}\right)}{2}\right)}{n} == \frac{\sum_{i=1}^{n}\left(T_{sp,pred,i}\right)}{n} \tag{5}$$

Table 4.1. Calculated Set Points Based on an Example of a Zone Temperature Profile

i	T_{max} (°F)	T_{min} (°F)	$T_{sp,pred}$ (°F)
1	71.9	71.5	71.7
2	71.8	71.5	71.6
3	72.0	71.5	71.7
4	71.9	71.5	71.7
5	71.9	71.5	71.7

4.3 Classification-Based Bayesian Classifier

When a building is occupied, the thermostat controls the RTU to maintain the zone temperature within a desired comfort level. When a building is unoccupied, the programmable thermostat allows the zone temperature to deviate from the occupied set point to an unoccupied set point, called night setup/setback. Therefore, the set point detection should identify all possible set point values.

A statistical classifier can distinguish the different set point temperatures more effectively. Several possible classifiers identify to which set of categories (e.g., occupied cooling set point and occupied heating set point, etc.) a new set point belongs. A Bayesian classifier is optimal with respect to minimizing the classification error associated with current and new normal distributions (Fukunaga 1990). Based on Bayes decision theorem, Equation (6) can be written as follows:

$$p(\omega_i \mid x) = \frac{p(x \mid \omega_i)\, p(\omega_i)}{p(x)} \tag{6}$$

where $p(\omega i|x)$ = the conditional probability of ωi having accounted for evidence x,
$p(\omega i)$ = prior probability,
$p(x|\omega i)$ = the class conditional probability of x, and
$p(x)$ = the mixture density function.

Because $p(x)$ is positive and common to both sides of the inequality, the Bayes decision rule of Equation (6) can be expressed as

$$\omega_1\,(Current\;) : p(x \mid \omega_1)\, p(\omega_1) \geq\; p(x \mid \omega_2)\, p(\omega_2) \tag{7}$$

$$\omega_2\,(New):\ p(x|\omega_1)\,p(\omega_1) \le\ p(x|\omega_2)\,p(\omega_2)$$

Figure 4.2 shows two conditional probabilities $P(x|\omega_i)$, $i = 1, 2$, as functions of x in each of the classes. The dashed line at x_0 is a threshold partitioned into two regions, R_1 and R_2. According to the Bayesian decision rule, for all x values in R_1 the classifier decides ω_1 and for all x values in R_2 it decides ω_2. However, there is overlapping probability, which is equal to the total shaded area under the curves belonging in R_1 and R_2, shown in Figure 4.2. The shaded area is the Bayesian classification error probability (ε), which is given by

$$\varepsilon = \int_{x_o}^{\infty} p(x|\omega_1)\,p(\omega_1)\,dx + \int_{-\infty}^{x_o} p(x|\omega_2)\,p(\omega_2)\,dx = p(\omega_1)\varepsilon_1 + p(\omega_2)\varepsilon_2 \qquad (8)$$

One of the most common probability density functions in practice is the normal probability density function. When $P(x|\omega_i)$ is the normal distribution with mean μ_i and covariance Σ_i, shown in Equation (9):

$$p(x|\omega_1) \sim N(\mu_1, \Sigma_1), \qquad p(x|\omega_2) \sim N(\mu_2, \Sigma_2) \qquad (9)$$

where
- x = a vector of current residuals,
- μ_1 = the mean describing the distribution of new set point,
- Σ_1 = the covariance describing the uncertainty of new set point,
- μ_2 = the mean describing the distribution of current set point, and
- Σ_2 = the covariance describing the uncertainty of current set point.

The Bayesian classification error is calculated by integrating the overlap area between probability distributions that fall within each class region of the domain, shown in Figure 4.2. The classification error probability decreases as an error of estimated value becomes more significant and is therefore a useful measure for distinguishing the current set point from new set point temperature. The thresholds for the classification error were established by evaluating the statistical significance of a match or mismatch between the current and the new set point temperature. For this research, the threshold was determined to be 0.1 classification error using existing field data.

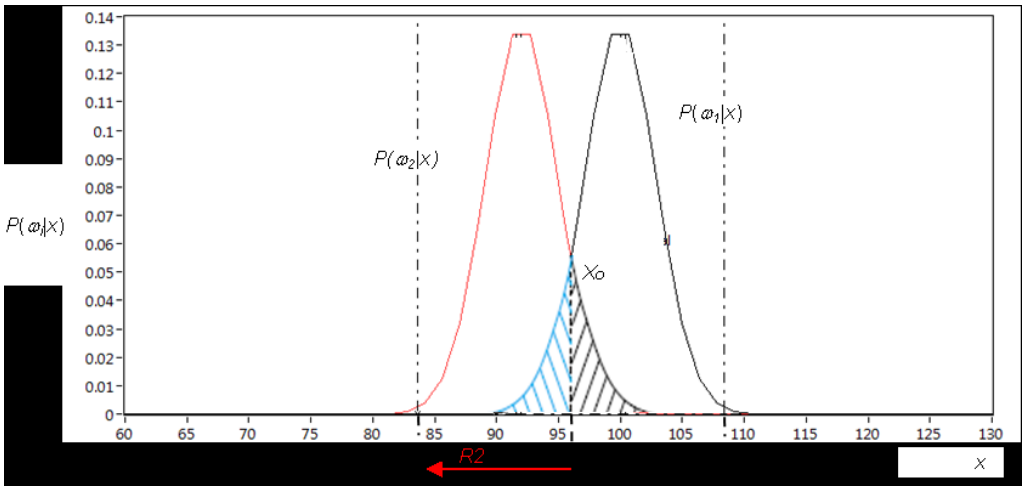

Figure 4.2. Bayesian Decision Rule for Minimum Error (ω_1: new set point and ω_2: current set point)

4.4 Metric Used for Zone Set Point Temperature Detection

In this section the metric used for the zone set point temperature detection is presented. According to International Standards Organization BS ISO 5725-1 (1994), the accuracy consists of both "trueness" and "precision." "Trueness" refers to the closeness of agreement between the mean of a "large" number of test results and the reference value, and "precision" refers to the closeness of agreement between test results, as shown in Figure 4.3. The confidence interval with *t*-distribution is used to quantify the accuracy of the set point detection. The width of the interval depends upon the confidence level and the precision of the prediction as shown in Equation (10).

$$\bar{x} - t_c \cdot \frac{s}{\sqrt{n}} < \mu < \bar{x} + t_c \cdot \frac{s}{\sqrt{n}} \tag{10}$$

where t_c = t-value for corresponding confidence level,
$\bar{x}$ = a mean of prediction,
n = a number of predictions, and
s = a standard deviation of prediction.

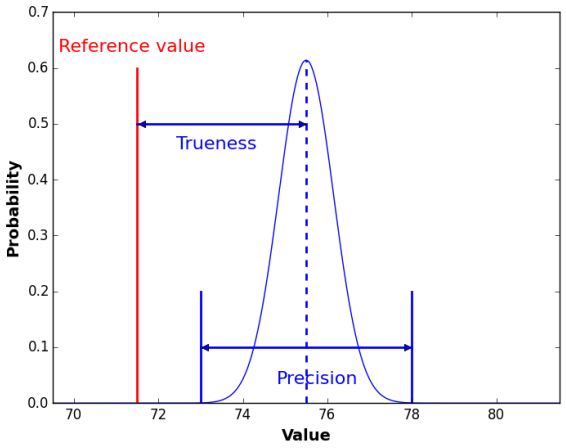

Figure 4.3. Definition of Accuracy (ISO 5725-1)

The confidence level is used as an accuracy/validation metric for measuring the success of the algorithm. The confidence level is the probability of a set point prediction being within a confidence interval (Altman et al. 2013). Depending on the confidence level chosen, the interval margin of error and respective range also change. The difference between the identified set point and the actual set point is used as a residual input to identify an interval wherein the prediction will lie. The goal of the algorithm is to identify the set point temperature +/-1.0°F and with a 90% confidence level.

4.5 Validation of Zone Set Point Detection Algorithm

The set point detection approach is explained using data from RTU-1 in Building B during the Spring/Fall-Day2. Figure 4.4 shows zone temperature and filtered zone temperature profiles. The blue and green lines indicate the raw zone temperature, and filtered zone temperature, respectively. The noise in raw zone temperature (blue) is reduced, while the peak is remains the same in the filtered zone temperature (green), making it easier to measure the peak position, height, and width.

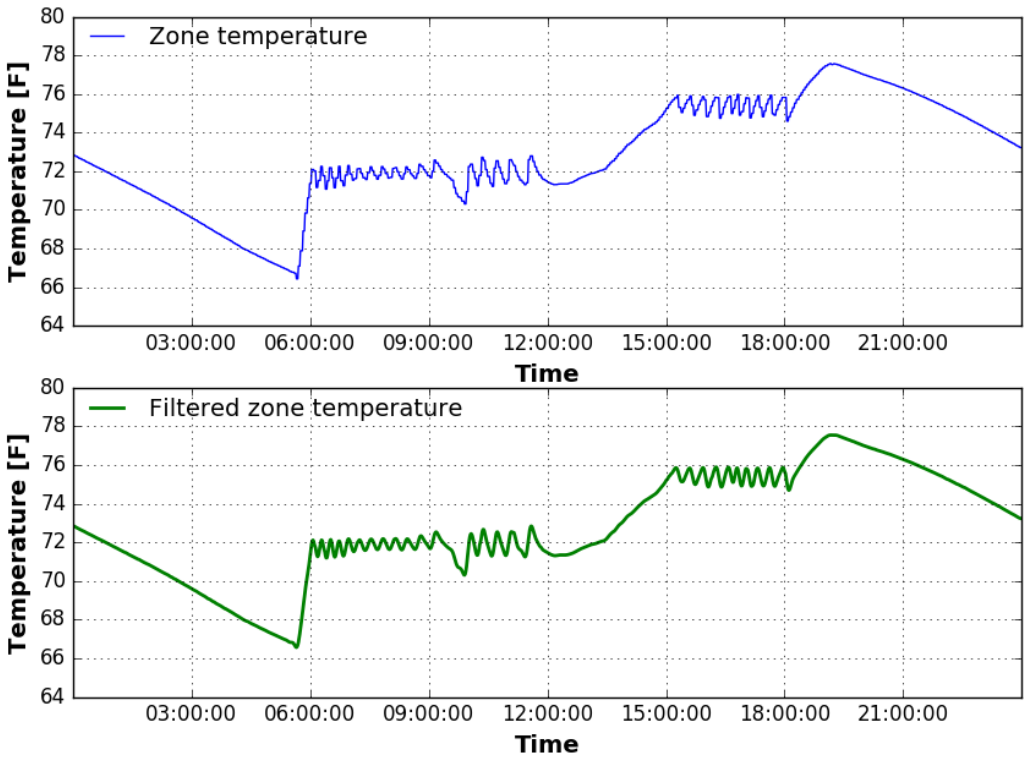

Figure 4.4. (a) Zone Temperature (top) and (b) Filtered Zone Temperature (bottom) Using Data from Building B-B-RTU1-Spring/Fall-Day 2

The first step in the set point detection process is to identify the set point or set points (for zones with multiple set points) in a 24-hour period (one day). Figure 4.5 shows zone temperature (blue) and the detected set point temperatures (green) over a 24-hour period. In this example, the zone has two distinct temperature set points: the first is the occupied cooling temperature set point (71.5°F) and the second is the unoccupied cooling temperature set point (75°F). The blue and red circles indicate T_{max} (blue) and T_{min} (red) estimated by set point detection algorithm. The total number of T_{max} and T_{min} points were 28 for this day. The T_{sp} was calculated using the average of corresponding T_{max} and T_{min}. The set point detection algorithm uses a fixed-length moving window average that creates series of average set point temperatures of different subsets of the full T_{sp} set. The size of subsets used for validation of the algorithm was 5 points. The first set point temperature was calculated by averaging the first five points from the T_{sp}, as shown in Figure 4.5. Then the subset was modified by excluding the first point from T_{sp} and adding the sixth point from T_{sp}. The average for points 2 to 6 represents the second set point temperature. This process is repeated over the full T_{sp} set. In this example, there will be 24 average set point temperatures (given that there are 28 points).

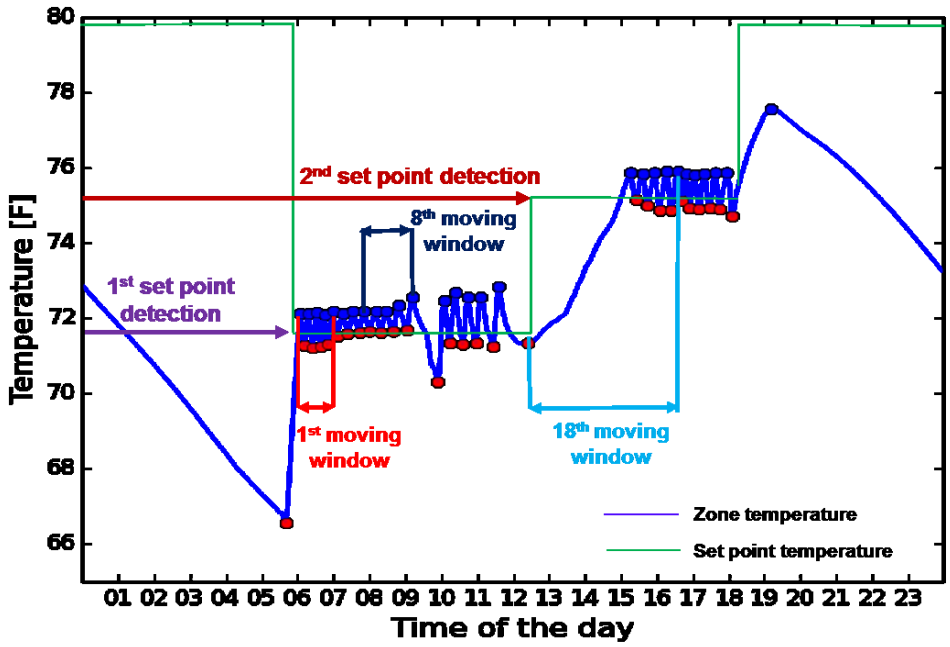

Figure 4.5. Zone Temperature and Cooling Command (Building B-RTU-1-Spring/Fall-Day 2)

The second step in the set point detection process, is to distinguish whether two adjacent detected set points are distinct and different. The probability distribution function is used to isolate distinct set points. Figure 4.6 shows an example of the probability distributions for two set points that correspond to the first and the eighth moving window subset. The mean for the first point (red dashed line) and eighth set point temperature (blue dashed line) are 71.9°F and 71.8°F, respectively. As discussed previously, the Bayesian classifier is used to estimate the classification error (overlap area) between the first and eighth distributions. The significant overlap between the first and eighth probability distributions indicates that there is no difference between the two sets. The classification errors based on the residuals between the first and eight set point is 0.59, indicating that there is no significant statistical difference between the two set point temperatures.

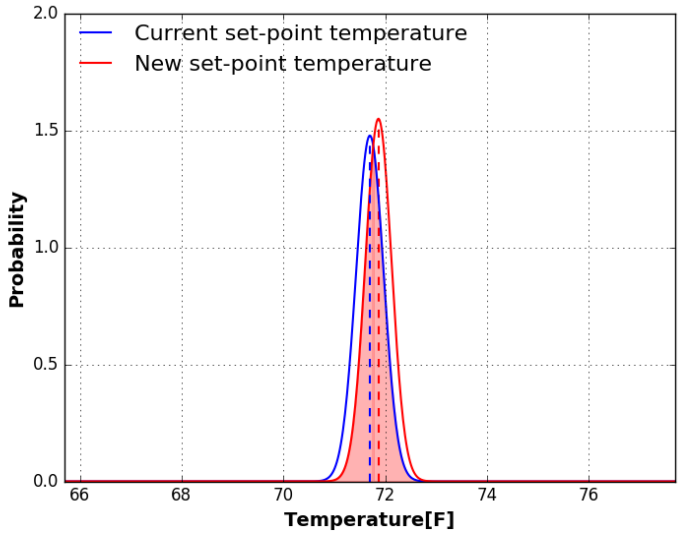

Figure 4.6. Overlapping Normal Distribution between the First and Second (new) Set Point (Building B-RTU1-Spring/Fall-Day 2)

The third and the final step in the set point detection process is to compare the estimated set point with the ground truth. As discussed in the previous section, the confidence level is used as an accuracy metric for measuring the success of set point detection. The difference between the predicted and the actual set point is used as a residual input to the confidence interval to identify an interval in which the prediction will lie. As stated earlier, the goal of the set point algorithm is to identify the set point temperature within ±1.0°F at a 90% confidence level. Figure 4.7 shows the probability density function. Using a sample data set, the algorithm identified the set point within 0.5°F of the actual set point. The confidence interval for this sample data set at a 90% confidence level is between -0.09°F and 0.45°F. Because the upper bound (0.45°F) for the sample data set is smaller than 1.0°F, the identified set point is considered to be correct with 90% confidence.

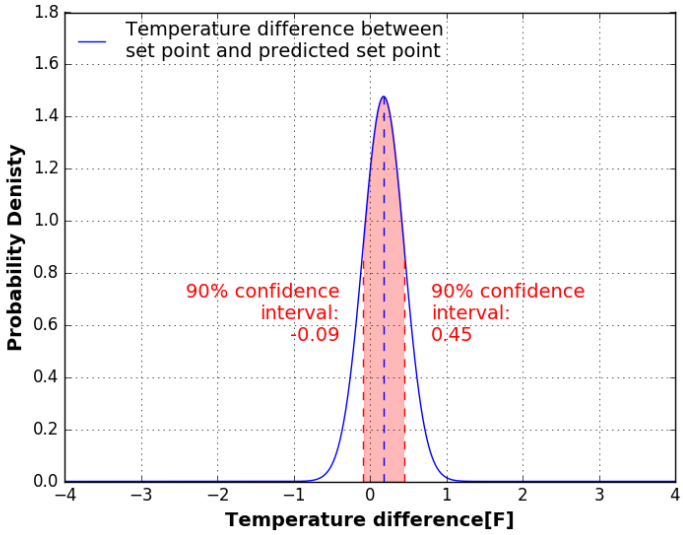

Figure 4.7. Probability Density Function of Correct Set Point Detetection (Building B-RTU2-Summer-Day 2)

Now that the first and the eighth set points were considered to be the same and also met the accuracy metric, the next step is to compare the distribution of the first and the ninth set points. This process of comparing the distributions continues for all set points. Figure 4.8 compares the first and the eighteenth moving window subsets, and shows much larger difference between the first and the eighteenth set point probability distributions, indicating that the set points are distinct and different. The Bayesian error between two set points (first and eighth) is 0.07, which is less than the established threshold (0.1).

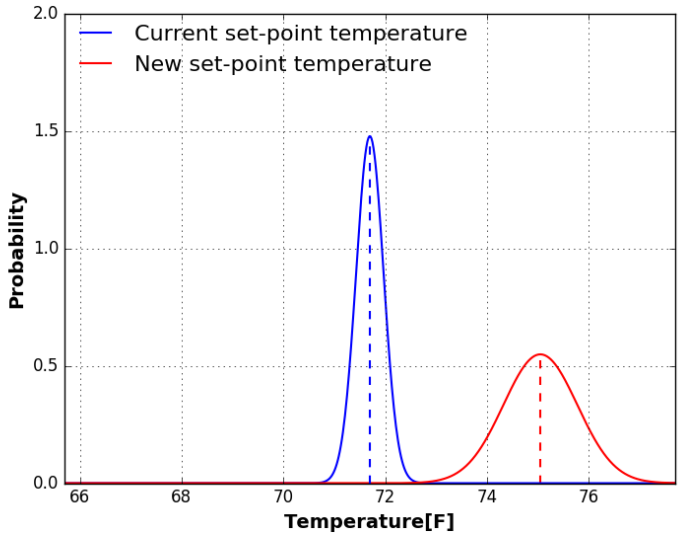

Figure 4.8. Overlapping Normal Distribution between the Second and the Third Set Point Temperatures (Building B-RTU1-Spring/Fall-Day 2)

Figure 4.9 shows the probability density function for the second distinct set point. The absolute difference between the predicted (75.1°F) and the actual (75.0°F) set point value is 0.1°F. The interval corresponding to the 90% confidence level lies between 0.2°F and 0.9°F. Because the upper bound of the confidence interval (0.9°F) is less than 1.0°F, the predicted set point is considered to be accurate to within 90% confidence.

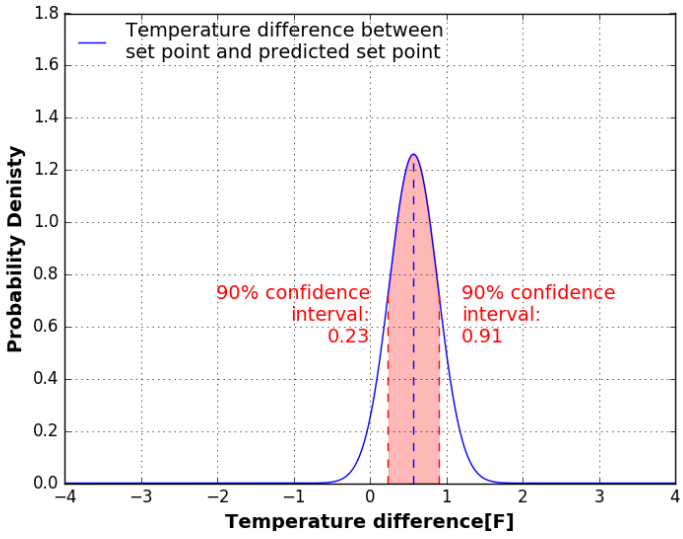

Figure 4.9. Probability Density Function for Set Point Detection (Building B-RTU-1-Spring/Fall-Day2)

Error! Reference source not found. shows the summary of the Building A data set used for validation of the set point detection algorithm and results. The evaluation was performed with a sample data from four RTUs spanning seven days of each season (summer, winter, and spring/fall) from four RTUs. The analysis included a total of 66 detections, including 61 correct and five incorrect detections. On 23 occasions no detection was possible because the number of ON/OFF cycles were less than five. Because the RTUs were OFF during the weekend, most of the "no" detections were from the weekend data.

Table 4.2. Summary of the Building A Data Set Used for Set Point Detection and Results

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection	Number of Incorrect Detection	Number of "NO" Detection
A	Seattle, WA	4	Summer	77	23	5	4
			Winter	43	18	0	10
			Spring/Fall	53	20	0	9
Total number of detections					61	5	23

Table 4.3 shows the summary of the Building B data set used for validation of the set point detection algorithm and results. The set point detection algorithm was evaluated using sample data from six RTUs spanning seven days for each season. The analysis included a total of 96 detections, including 88 correct and 8 incorrect detections. Six of the eight incorrect detections happened in spring/fall. Although the units cycled ON/OFF more than five times, it was not frequent enough to make correct detections. On 44 occasions no detection was possible because the number of ON/OFF cycles were less than five and most of these days were from the weekend data.

Table 4.3. Summary of the Building B Data Set Used for Set Point Detection and Results

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection	Number of Incorrect Detection	Number of "NO" Detection
B	Berkeley, CA	6	Summer	70	31	2	14
			Winter	51	26	0	16
			Spring/Fall	64	31	6	14
Total number of detections					88	8	44

Table 4.4 shows the summary of the Building C data set used for validation of the set point detection algorithm and results. The set point detection algorithm was evaluated using sample data from eight RTUs spanning seven days for each season. The analysis included a total of 116 detections, including 95 correct and 21 incorrect detections. The number of incorrect detections for this building was relatively higher than for the other two buildings, and the reason for this is explained below (Figure 4.10). On 60 occasions no detections were possible because the number of ON/OFF cycles were less than five; this situation occurred mostly during weekends. The number of "no" detections was also higher for this building than for the other two buildings because this building is open space and thus only some of eight RTUs were necessary to maintain the comfort in the space.

Table 4.4. Summary of the Building C Data Set Used for Set Point Detection and Results

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection	Number of Incorrect Detection	Number of "NO" Detection
C	Richland, WA	8	Summer	73	33	10	15
			Winter	41	25	4	28
			Spring/Fall	63	37	7	17
Total number of detections					95	21	60

Figure 4.10 shows the zone temperature profile and cooling command for the HP4 system in Building C during the cooling season. The blue, red, and black lines represent the zone temperature, set point temperature, and the cooling command signal, respectively. Regarding the 21 incorrect detections, in some cases the system was cycling before it reached the set point temperature of 70°F; therefore, the predicted set point value was higher than the actual zone set point temperature.

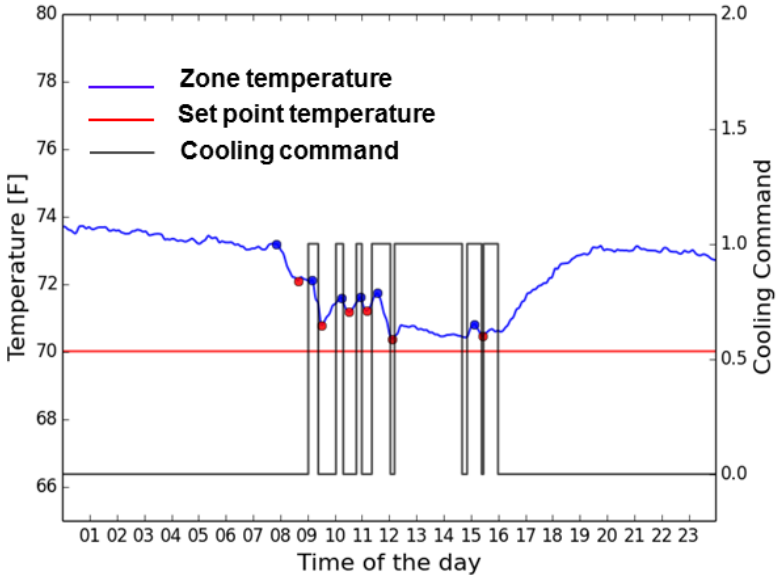

Figure 4.10. Temperature Profile and Cooling Command (Building C-HP4-Summer/Day2)

Table 4.5 shows the summary of the Building C data set used for validation of the set point detection algorithm and results. No winter data are used because both RTUs are air conditioners only. The set point detection algorithm was evaluated using sample data from two RTUs spanning seven days. The analysis included 22 correct and 1 incorrect detection and on 5 occasions detection was not possible. During summer, all set points were detected correctly with one exception, because the building was a grocery store with a single set point all week.

Table 4.5. Summary of the Building D Data Set and Results for Set Point Detection

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection	Number of Incorrect Detection	Number of "NO" Detection
D	South Paris, ME	2	Summer	73	13	1	0
			Spring/Fall	60	9	0	5
Total number of detections					22	1	5

Table 4.6 shows the summary of the Building E data set for validation of the set point detection algorithm and results. All set points were correctly detected with one exception.

Table 4.6. Summary of the Building E Data Set and Results for Set Point Detection

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection	Number of Incorrect Detection	Number of "NO" Detection
E	Miami, FL	2	Summer	84	13	1	0
			Winter	69	14	0	0
			Spring/Fall	80	14	0	0
Total number of detections					41	1	0

Table 4.7 shows the summary of Building F for validation of the set point detection algorithm and results. The analysis included 37 detections—29 correct and 8 incorrect. Five set points could not be detected because the building has a single set point (Figure 4.10).

Table 4.7. Summary of the Building F Data Set and Results for Set Point Detection

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection	Number of Incorrect Detection	Number of "NO" Detection
F	Cupertino, CA	2	Summer	73	12	1	1
			Winter	50	9	4	1
			Spring/Fall	61	8	3	3
Total number of detections					29	8	5

Table 4.8 shows the validation summary for the set point detection algorithm. The algorithm correctly detected 354 out of 399 set point temperatures. The details of the results for each RTU in three of the buildings are presented in **Error! Reference source not found.**. Although the absolute difference between the predicted and actual set point temperature for 45 cases was greater than 1.0°F, the difference between the predicted and actual value was within ±1.5°F (see red shaded data in **Error! Reference source not found.**) for most of these cases. Because the set point detection algorithm requires a minimum number of peaks and valleys to predict set point temperature, it had difficulty when the number of peaks and valleys was less than five (see gray shaded data in **Error! Reference source not found.**) or relatively small. Overall, the performance of the algorithm was good; it estimated the set point accurately to within 1°F 88% of the time and within 1.5°F 100% of the time.

Table 4.8. Summary of the Validation Results for the Set Point Detection Algorithm

Total Number of Set Point Temperatures Detected	Total Number of correct Set point temperatures detected	Total Number of Incorrect Set Point Temperatures Detected
390	356	45

The set point detection algorithm has been shown to provide accurate set point temperature estimates using just the zone temperature data from a number of RTUs in diverse climate locations and building types. The set point detection algorithm can be deployed on SMBs that do not have BASs using a low-cost single board computers (e.g., Raspberry PI, BeagleBone, etc.) with a single inexpensive temperature sensor to measure zone temperature. The algorithm can generate actionable information, including detecting incorrect temperature set points (set points changed by someone) or an override of the thermostat.

5.0 Development and Validation of ON/OFF Cycling Detection Algorithm

This section describes the methodology used to detect RTU ON/OFF cycling based on zone temperature measurement. The RTU can provide mechanical cooling (direct expansion vapor compression), from cool outside air (economizing) or a combination of mechanical cooling and economizing (when economizing alone cannot meet the cooling needs of the conditioned zone). The ON/OFF cycling algorithm is useful when the thermostat's cooling command value is not available. ON/OFF cycling detection can be used to quantify the number of times an RTU is ON or OFF in a given time period. The number of detected cycles being higher than a predefined threshold during a certain time period indicates an RTU short cycling problem. RTU cycling is a function of a number of variables, including oversizing and temperature differences between the indoors and outdoors. The RTU is expected to cycle more often when the temperature difference between indoors and outdoors is high or when the unit is significantly oversized.

5.1 Development of ON/OFF Cycling Detection

In a zone temperature profile, the peak indicates "ON" and the valley indicates "OFF." The peak detection (Peak Detection Algorithm) can detect the number of peaks (ON)/valleys (OFF) in the daily zone temperature time-series data. The algorithm reads T_{zone}, searches for valid peaks and valleys, and keeps track of the total number of peaks and valleys found. The overall number of ON/OFF cycles in a day can be calculated by the sum of the number of peaks. The algorithm can detect excessive equipment ON/OFF cycling and equipment that remains in an ON or OFF state for significant periods of time. The technique does not require any supervised learning or extra additional sensor installation, which is costly and time consuming. Two thresholds, which are user adjustable configuration parameters, are described below:

5.1.1 Maximum Number of Cycles

The maximum cycling number represents the maximum number of cycles expected per day. A cycling problem is detected when the number of cycles exceeds the maximum cycling number (default value 100). If the number of detected cycles is higher than a predefined threshold during a certain time period, it indicates an RTU is short cycling.

5.1.2 Minimum Number of Cycles

The minimum cycling number represents the minimum number of cycles per day. A cycling problem is detected when an RTU remains either in the ON or OFF state for significant period of time (default minimum is value 0).

5.2 Metric Used for ON/OFF Cycling Detection

This section describes the metric used for the ON/OFF cycling detection. The mean absolute percent error (MAPE) estimate is used to quantify the accuracy of the ON/OFF cycling detection algorithm. The MAPE between identified and actual value is a widely used accuracy metric. The MAPE can be calculated as the average absolute percent error, as shown in Equation (11). The ON/OFF cycling algorithm estimates the number of ON/OFF cycles over the 24-hour period. The goal of the ON/OFF

cycling detection algorithm is to identify the number of ON/OFF cycles with a MAPE value of less than 20%.

$$MAPE\ metrics\ (\%) = \left(1 - \frac{1}{n}\sum_{i=1}^{i=n}\left|\frac{Actual_i - Identified_i}{Actual_i}\right|\right) \times 100 \quad (11)$$

The MAPE is scale sensitive and should not be used when working with limited amount of test data. Because "Actual" is in the denominator of Equation (11), the MAPE is undefined when "Actual" is quite small, the MAPE will often take on extreme values.

5.3 Validation of ON/OFF Cycling Detection

First, the cycling detection algorithm is explained using cooling season data from RTU-1 in Building D. Figure 5.1 shows the zone temperature profile and cooling command for RTU-1 in Building D during the cooling season. The blue and black lines indicate the zone temperature and compressor signal, respectively. As seen in the figure, the T_{max} and T_{min} correctly correspond to compressor signals 0 (OFF) and 1 (ON), respectively. The total number of cycles predicted by the algorithm was 35/day and the actual number of cycles was 34/day. The corresponding MAPE metric is 97%, indicating an accurate prediction of ON/OFF cycles.

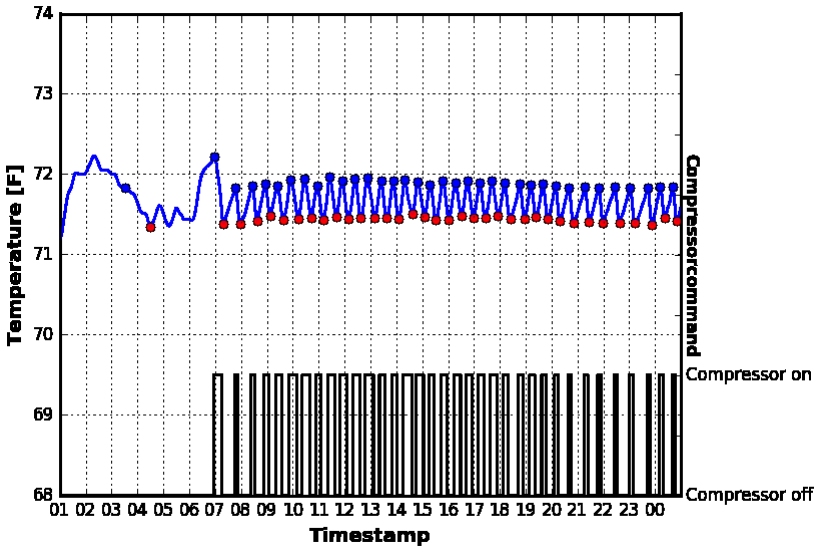

Figure 5.1. Zone Temperature Profile and Cooling Command (Building D-RTU1-Summer/Day 1)

Table 5.1 shows the summary of the Building A data set used for validation of the ON/OFF cycling detection algorithm and results. The results are split into two groups: 1) a group of days when the ON/OFF cycles numbered more than 15 events and 2) a group of days when the ON/OFF cycles numbered more than 5. For days when the ON/OFF cycles exceeded 15, the algorithm was more successful in detecting the ON/OFF cycles than when the ON/OFF cycles exceeded 5.

Table 5.1. Summary of the Building A Data Set and Results for the ON/OFF Cycling Detection

Location	Season	$T_{OA,AVG}$ °F	Daily ON/OFF Cycles Greater than 5		Daily ON/OFF Cycles Greater than 15	
			Number of Correct Detection	Number of Incorrect Detection	Number of Correct Detection	Number of Incorrect Detection
Seattle, WA	Summer	77	16	7	9	3
	Winter	43	14	4	9	1
	Spring/Fall	53	18	3	10	0
Total number of detections			48	14	28	4
% of correct/incorrect detection			77%	23%	88%	12%

Table 5.2 shows the summary of Building B data set used for the validation of the ON/OFF cycling detection algorithm and results. The data with more than 5 daily ON/OFF cycles were 80% accurate. The accuracy increased to 85% when the data only included days with 15 or more ON/OFF cycles.

Table 5.2. Summary of the Building B Data Set and Results for the ON/OFF Cycling Detection

Location	Season	$T_{OA,AVG}$ °F	Daily ON/OFF Cycles Greater than 5		Daily ON/OFF Cycles Greater than 15	
			Number of Correct Detection	Number of Incorrect Detection	Number of Correct Detection	Number of Incorrect Detection
Berkeley, CA	Summer	70	26	6	16	3
	Winter	51	17	4	17	3
	Spring/Fall	64	15	5	12	2
Total number of detections			58	15	45	8
% of correct/incorrect detection			80%	20%	85%	15%

Table 5.3 shows the summary of the Building C data set used for validation of the ON/OFF cycling detection algorithm and results. The data with more than 5 daily ON/OFF cycles was 65% accurate, while the accuracy increased to 88% when only data with 15 or more cycles were used. The incorrect detections are explained in Figure 5.2.

Table 5.3. Summary of the Building C Data Set and Results for the ON/OFF Cycling Detection

Location	Season	$T_{OA,AVG}$ °F	Daily ON/OFF Cycles Greater than 5		Daily ON/OFF Cycles Greater than 15	
			Number of Correct Detection	Number of Incorrect Detection	Number of Correct Detection	Number of Incorrect Detection
Richland, WA	Summer	70	32	18	22	4
	Winter	51	21	9	10	2
	Spring/Fall	64	28	17	18	1
The total number of detections			81	44	50	7
% of correct/incorrect detection			65%	35%	88%	12%

Figure 5.2 shows the zone temperature profile and cooling command of the HP3 system in Building C during the cooling season. The blue and red lines indicate the zone temperature and cooling command signal, respectively. As shown, the cooling command signals are not corresponding to the zone temperature trend. This is because several RTUs in Building C serve open space. Therefore, unknown but

highly possible inter-zonal convective coupling is occurring due to close interactions between the different RTUs and their thermostats, non-uniformly distributed diffusers, and non-uniform heat gains.

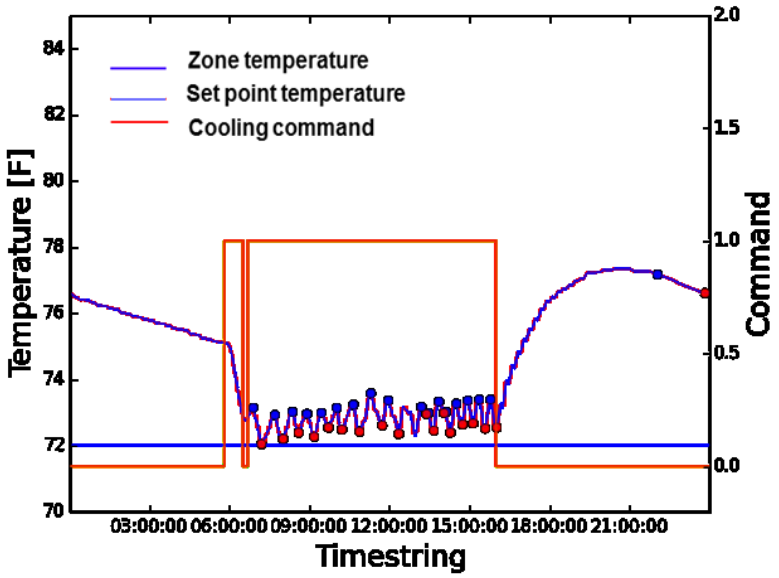

Figure 5.2. Temperature Profile and Cooling Command (Building C-HP3-Summer/Day 2)

Table 5.4 shows the summary of the Building D data set used for validation of the ON/OFF cycling detection algorithm and results. The data with more than 5 daily ON/OFF cycles were 87% accurate, while the accuracy increased to 100% when only data with 15 or more cycles were used.

Table 5.4. Summary of ON/OFF Cycling Detection Results for Building D

			Daily ON/OFF Cycles Greater than 5		Daily ON/OFF Cycles Greater than 15	
Location	Season	$T_{OA,AVG}$ °F	Number of Correct Detection	Number of Incorrect Detection	Number of Correct Detection	Number of Incorrect Detection
South	Summer	73	14	0	12	0
Paris, ME	Spring/Fall	60	6	3	2	0
Total number of detections			20	3	14	0
% of correct/incorrect detection			87%	13%	100%	0%

Excessive (short) cycling of the system can lead to premature failure of the compressor or its components. Short cycling also can result in a significantly decreased average efficiency, even if there are no physical failures in the equipment. According to the Architectural Energy Corporation's Small HVAC System Design Guide (Jacobs 2003b), short cycling can cause an efficiency penalty of approximately 10%. Table 5.5 shows the summary results of the ON/OFF cycling detection algorithm. For data with more than five ON/OFF cycles, the algorithm detected 203 out of 283 cycles correctly. For data with more than 15 ON/OFF cycles, the algorithm detected 133 out of 156 cycles correctly. Because the goal is to detect frequent ON/OFF cycles, lower accuracy of detection when there are fewer cycles than 15 is not critical. The details of the results for each RTU in three the buildings are presented in

Validation of On/Off Cycle Algorith.

Table 5.5. Summary of Daily ON/OFF Cycle Detection Results

Daily ON/OFF Cycles Greater than 5 (A total of 359)		Daily ON/OFF Cycles Greater than 15 (A total of 196)	
Number of Correct ON/OFF Cycles Detection (>80%)	Number of Incorrect ON/OFF Cycles Detection (<80%)	Number of Correct ON/OFF Cycles Detection (>80%)	Number of Incorrect ON/OFF Cycles Detection (<80%)
203	80	133	23

Table 5.6 shows the summary of weekly ON/OFF cycling detection. The weekly MAPE was calculated based on a sum of cycling over a 1-week period. For data with more than 75 weekly ON/OFF cycles, the algorithm was incorrect only once.

Table 5.6. Summary of Weekly ON/OFF Cycle Detection Results

Weekly ON/OFF Cycles Greater than 25 (A total of 64)		Weekly ON/OFF Cycles Greater than 75 (A total of 47)	
Number of Correct ON/OFF Cycles Detection (>80%)	Number of Incorrect ON/OFF Cycles Detection (<80%)	Number of Correct ON/OFF Cycles Detection (>80%)	Number of Incorrect ON/OFF Cycles Detection (<80%)
49	5	30	1

6.0 Development and Validation of Occupancy Schedule Detection Algorithm

The goal was to develop, implement, and validate methods for estimating the occupancy schedule without additional sensor requirements. The building operator may have difficulty trying to validate the occupancy schedules based upon zone temperature history charts without other measurements (e.g., supply fan signal). The method reported here for detecting the occupancy schedule is based only on zone temperature measurement from continuous time-series data. To detect the occupancy information from the zone temperature, we use SAX to detect the occupancy schedule from a daily zone temperature profile. The detected schedule can help the building operator identify the occupancy schedule patterns and the changes in or overrides of the schedule. Also, the scheduling information derived from this algorithm can be used to optimize the operating schedules (e.g., precooling, time-of-day scheduling) of RTUs to restore energy-efficient operation. To validate the performance of the algorithm, the predicted occupancy schedule was compared with actual occupancy information using a confusion matrix.

The development of occupancy schedule detection is broken down into three steps—data preprocessing, SAX transformation, and data clustering—designed to detect the daily schedule from zone temperature time series. In the data preprocessing step, the temperature data are normalized to a mean of zero and a standard deviation of one. Next, the normalized temperature is transformed into the SAX representation by creating groups of SAX words from daily windows. This approach can preserve meaningful patterns from the temperature data and produce competitive results for classifying schedules (Lin et al. 2003). The basic idea of SAX is to convert the time-series data into a discrete format with a small alphabet size. In this case, every part of the representation contributes about the same amount of information about the shape of the time series. The additional clustering step beyond the SAX transformation adds the ability to aggregate the daily occupancy profiles. The field test data (summer season data from RTU-3 at Building A) are used to describe the development of the schedule detection algorithm.

6.1 Data Preprocessing

In the data mining approach, data preprocessing is an important step for cleaning and standardizing the data (Goldin and Kanellakis 1995). In this effort, the temperature data are normalized to use the Gaussian distribution as the data model. The normalized temperature is calculated by subtracting the mean and dividing the standard deviation as show in Equation (12).

$$T_n(t) = \frac{T_{zone}(t) - \mu_X}{\sigma_X} \tag{12}$$

where Tn(t) = the normalized temperature time series of length n,
Tzone (t) = the temperature time series of length n,
μ_X = the mean of Tzone (t), and
σ_X = the standard deviation of Tzone (t).

In the current algorithm, *n* is 1440 (24 hours of 1-minute frequency data). Figure 6.1 shows the zone temperature and normalized zone temperatures of RTU-3 in Building A during the summer season. Each temperature time series was normalized, which does not affect the original shape of zone temperature measurements and scales the data to be comparable.

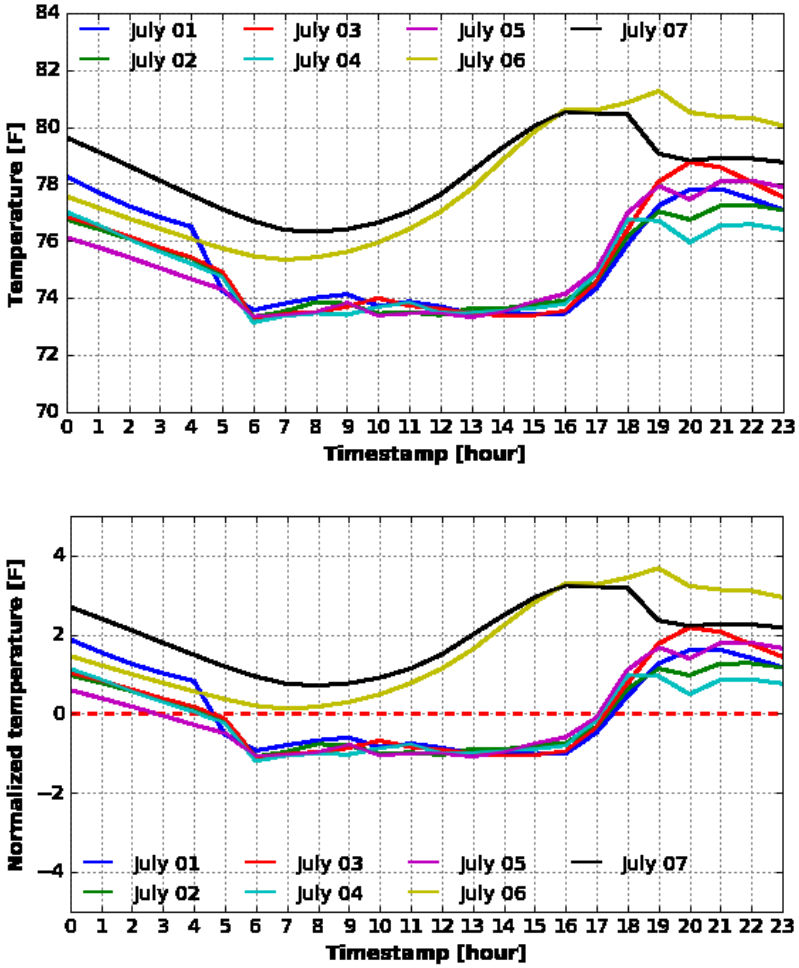

Figure 6.1. Zone Temperature and Normalized Zone Temperatures (Building A-RTU-3-Summer Season)

6.2 Data Mining Approaches Based on SAX Transformation

To convert a normalized temperature into SAX symbols, two steps of discretization are performed.

6.2.1 Piecewise Aggregate Approximation

First, the data are transformed into the Piecewise Aggregate Approximation (PAA) representation. The time series $T_n(t)$ of length n is divided into w equal-sized segments (typically $w << n$). In the current approach, w is chosen as a 1-hour segment due to the focus on hourly schedule characterization in buildings. The values in each segment are then approximated and replaced by the average of the data points in each segment (Keogh et al. 2000). The i^{th} element of $\bar{T}$ is calculated by Equation (13). Because w is typically much smaller than n, PAA representation can often largely reduce the dimensionality and makes the computation of the time-series data more efficient.

$$\bar{T}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}\cdot i} T_n(j) \tag{13}$$

Figure 6.2 shows the example result of PAA dimensional reduction of RTU-3 in Building A during the summer seasons. The blue and red lines indicate PAA representation and normalized temperature profiles. The normalized temperature time series is converted into PAA representation. This dimensionality reduction technique preserves the general shape of temperature data. To reduce the time series from 1440 dimensions (1-minute interval) to 24 equal size segment (1-hour interval), the temperature data are divided into 24 equal-sized segments. The temperature measurements in each segment are replaced by the average value of the data points.

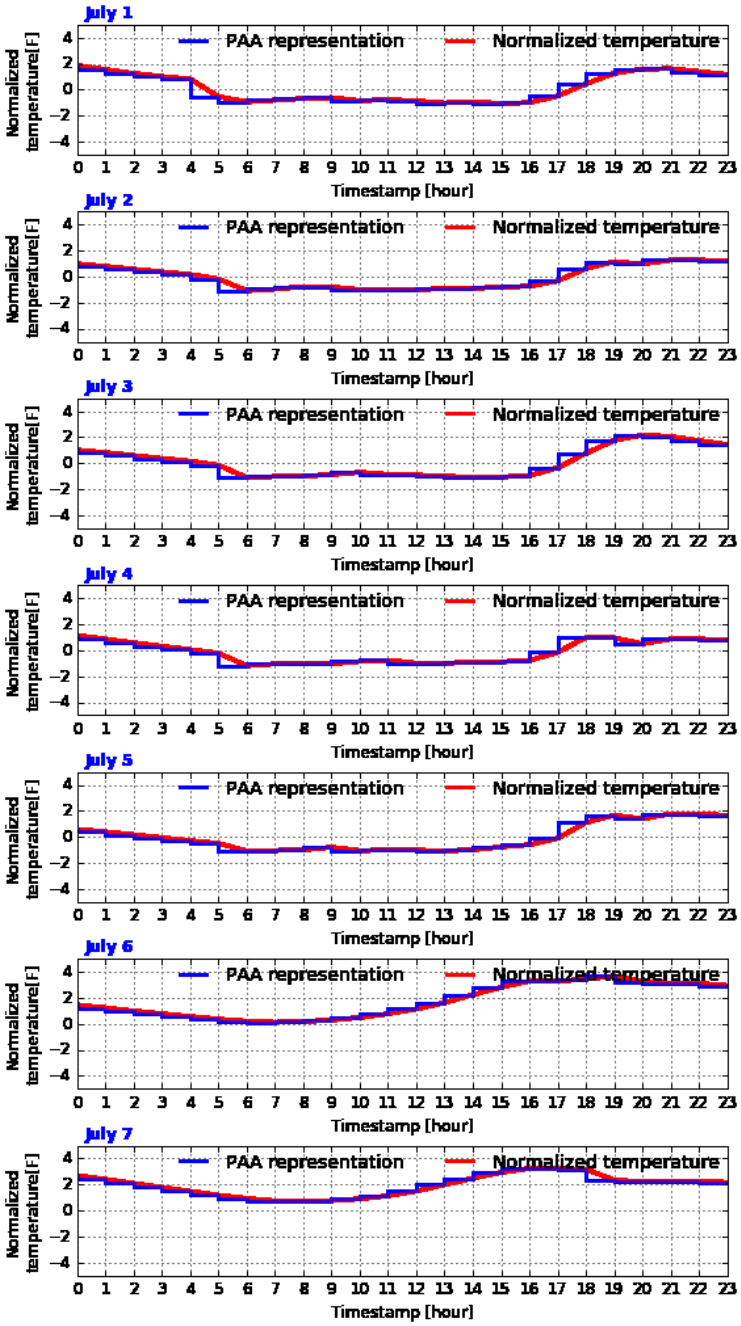

Figure 6.2. PAA Representation and Normalized Zone Temperature (Building A-RTU-3-Summer Season)

6.2.2 Discretization Based on Symbol

After having transformed the temperature time series into the PAA representation, an alphabetic symbol is assigned according to where the mean lies with a set of vertical breakpoints. These breakpoints are calculated according to a chosen number of alphabet size, *A* to create equi-probable regions based on a Gaussian distribution because the normalized time series have highly Gaussian distributions (Keogh and Lin 2005). Based on a chosen number of *A*, each PAA representation is transformed into SAX words. A

larger alphabet size creates SAX words with more diverse ranges of characters and captures more relative to resolution magnitude. For the current work, four alphabet sizes were selected to create three breakpoints (-0.67, 0, and 0.67) on the x-axis by looking them up in a statistical table. Three breakpoints can produce four equal-sized areas under the Gaussian curve as shown in in Figure 6.3. Three red dashed line indicate the breakpoints. Each region is represented by symbols A, B, C, and D.

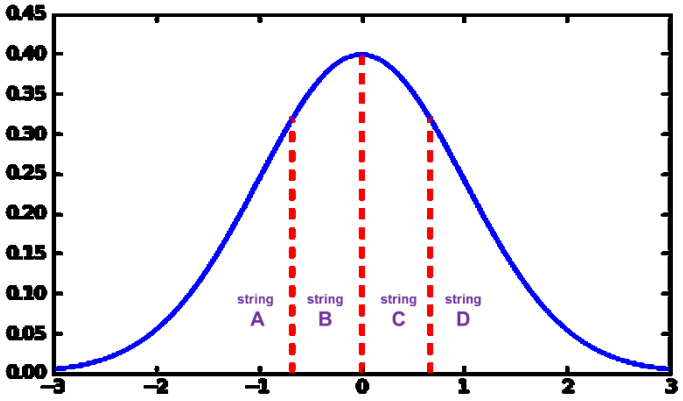

Figure 6.3. Breakpoint by Using the Gaussian Distribution

Figure 6.4 shows the frequency of the SAX strings of RTU-3 in Building A during the summer seasons. The blue and red lines indicate the SAX representation and normalized temperature for the example 1-week data. The normalized temperature is discretized by first obtaining a PAA representation and then using an alphabet string to map the PAA representations into SAX symbols. Each breakpoint is replaced by the string symbol A, B, C or D. For example, the PAA representations that are below -0.67 are mapped to the symbol D, PAA representation greater than or equal to -0.67 and less than 0 are mapped to the symbol C. In the example, for July 1, the zone temperate time series is mapped to the string, "DDDDDBAAABAAAAAAABCDD DDD." The SAX method can roughly preserve the general shape of the normalized zone temperature with large dimensionality reduction. The SAX string is helpful from an interpretative point of view in that each letter corresponds consistently to subsequent data from the daily temperature profile. For example, the string D describes the pattern for the hours of night. Therefore, a SAX word whose first letter is A would have low temperature, B and C would indicate relative average temperatures, and D would correspond to high temperature.

The key parameters of SAX representation creation are the w segment and alphabet size A. Decreasing the number of parameters creates a more simplified situation that should be considerably easier to interpret; however, the downside is the low level of detail. Increasing the number of parameters creates the most detailed patterns, which may be overwhelming in an analysis situation. To get the best values of the two parameters, we used the temperature time series from several buildings and made a search of patterns with several combinations of values. Based on these observations of the A and w, it was found that the setting $w = 24$ and $A = 3$ resulted in the best balance between the number of patterns generated and resolution of detail needed to adequately detect occupancy schedule in a 24-hour period. While these findings are specific to our case studies, we hypothesize that similar settings will be useful when analyzing other zone temperature data due to the generally reoccurring daily patterns. These initial parameter settings may be used as a default when implementing the occupancy schedule algorithm and adjusted accordingly based on the temperature profiles and measurement frequency.

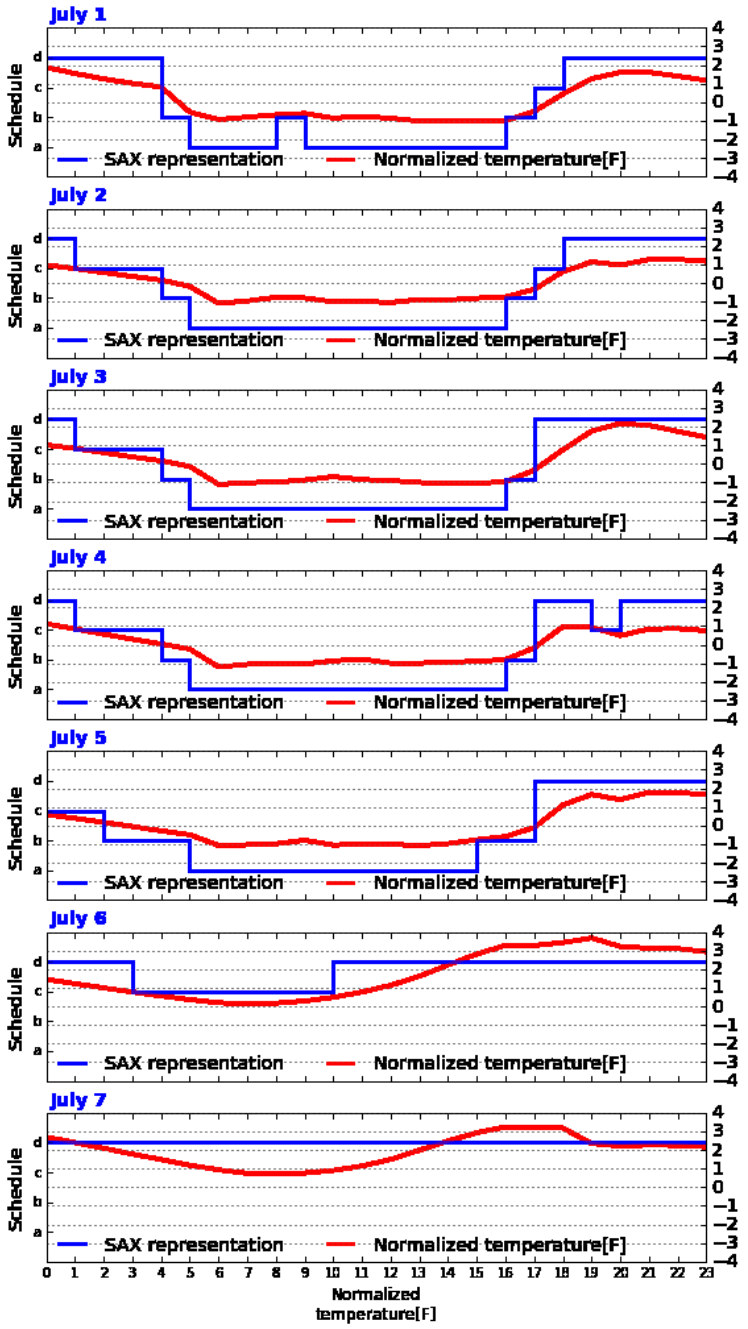

Figure 6.4. One-Week Sample SAX Representation (Building A-RTU-3-Summer Season)

Once the SAX words are created, the next step is to identify each occupancy schedule. The results of applying the occupancy process to RTU-3 data in Building A during the summer seasons are shown in Figure 6.5. The blue and red dashed lines indicate the predicted and actual occupancy schedule. The actual weekday occupancy schedule is from 5:00 a.m. to 4:00 p.m. and the building is not occupied over the weekend. The patterns between "Occupied schedule" and "Unoccupied schedule" can be distinguished according to SAX symbols. The SAX representations "A" and "B" are set to "Occupied schedule." Otherwise, the SAX representation C or D is set to "Unoccupied schedule." Based on this notation, the predicted weekday occupancy schedule for July 1 is 4:00 a.m. to 5:00 p.m. The SAX method was able to predict the occupied schedule.

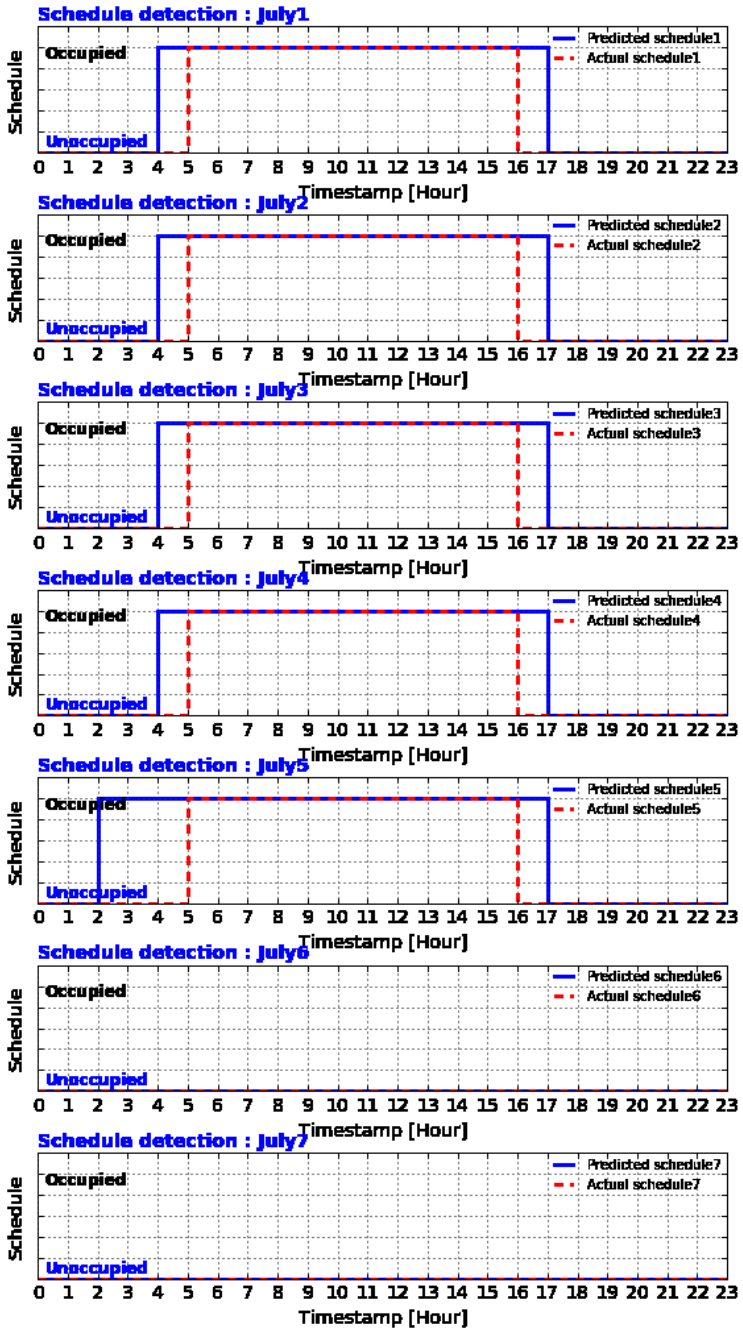

Figure 6.5. Comparison between the Actual and Predicted Schedule (Building A-RTU-3-Summer Season)

6.3 Clustering

After creating the occupancy profile candidates, the algorithm clusters the occupancy candidates to create possible schedules (e.g., weekend, weekday). The clustering step is supplementary if the SAX transformation process produces too many occupancy candidates (Lin et al. 2003). Clustering would be useful, for example, if seven occupancy candidates are created and the user wants to further aggregate those candidates into one or two more typical occupancy schedule(s).

For the current approach, a distance measure function, *MINDIST,* is used to cluster the daily occupancy profiles after creating the possible candidate. *MINDIST* is defined to identify patterns after transforming to SAX representation, as shown in Equation ((14). The distance between two SAX representations requires looking up the distances between each pair of symbols, squaring them, summing them, taking the square root, and finally multiplying by the square root of the compression rate. This function returns the minimum distance between the expected time series and its observed time series. For example, if two schedules are similar, then the distance measurement between their SAX representations is less than the target distance. In this approach, the target distance is set as 2.0 to distinguish two different schedules.

$$MINDIST(Case\ 1, Case\ 2) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} \left(dist(symbol\ 1, symbol\ 2)\right)^2} \tag{14}$$

where the *dist* function is implemented by using the lookup table for the particular set of the breakpoints as shown in Table 6.1. The distance between two symbols can be read off by examining the corresponding row and column. Entries for both A ↔ B and B ↔ C have identical values. For example, *dist*(A, B) = *dist*(B, A) = 0 and *dist*(A,C) = *dist*(C,A) = 0.67.

Table 6.1. Distance between Two Symbols

	A (=area 1)	B (=area 2)	C (=area 3)	D (=area 4)
A (=area 1)	0	0	0.67	1.34
B (=area 2)	0	0	0	0.67
C (=area 3)	0.67	0	0	0
D (=area 4)	1.34	0.67	0	0

Table 6.2 shows two daily profiles (Building A-RTU-3- July 1 and July 2) that are converted to the SAX words. The distance between the two SAX representations returns a value of 0 based on Table 6.2. The schedule extracted from July 1 is the same as the schedule from July 2.

Table 6.2. Example of Distance between Two SAX Representations (Building A-RTU-3- July 1 and July 2)

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
July.1st	d	d	d	d	d	b	b	b	b	b	b	b	b	a	b	a	a	b	d	d	d	d	d	d
	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
July.2nd	d	d	d	d	c	c	a	b	b	b	a	b	a	b	b	b	b	b	d	d	d	d	d	d
Distance	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.3 shows two daily profiles (Building A-RTU-3- July 1 and July 6) that are converted to the SAX words. When the distance between the two zone temperatures is higher than 2.0, the two time series should be considered the result of different schedules. Therefore, the schedule extracted from July 1 is different from that of July 6.

Table 6.4 shows the summary of distance results using RTU-3 data in Building A during the summer seasons. The distances between July 1 and all other days were calculated to create distinct groups of daily profiles. Two candidates appear to be the typical weekday and weekend schedule. Figure 6.6 shows two detected weekday schedules and weekend schedules for the example one-week data.

Table 6.3. Example of Distance between Two SAX Representations (Building A-RTU-3- July 1 and July6)

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
July.1st	d	d	d	d	d	b	b	b	b	b	b	b	B	a	b	a	a	b	d	d	d	d	d	d
	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
July.6th	d	d	d	d	d	d	c	c	c	c	d	d	D	d	d	d	d	d	d	d	d	d	d	d
Distance	0	0	0	0	0	0	0	0	0	0	.7	.7	.7	1.	.7	1.	1.	.7	0	0	0	0	0	0

Table 6.4. Summary of Distance Results (Building A-RTU-3-Summer Season)

	Weekday				Weekend	
	July 2	July 3	July 4	July 5	July 6	July 7
July 1st	0	0	0	0	2.86	3.16

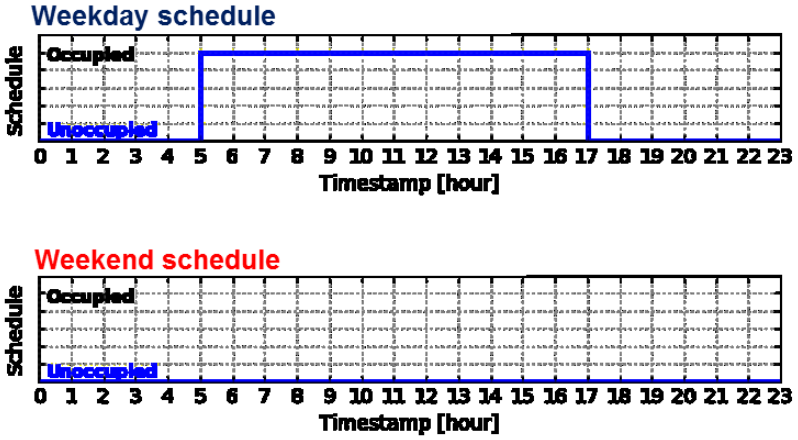

Figure 6.6. Weekday and Weekend Occupancy Schedule (Building A-RTU-3-Summer Season)

6.4 Metric Used for the Scheduling Detection Approach

This section describes the metric used for the detection of scheduling. A confusion matrix (Provost and Kohavi 1998) was used to quantify the accuracy of the schedule detection algorithm. The matrix shows the number of correct and incorrect predictions made by the algorithm compared with the actual occupied/unoccupied schedule in the test data. Table 6.5 shows the confusion matrix for the schedule detection algorithm. The rows and columns of the matrix correspond to the number of actual schedules in the test data and the number of predicted schedules made by the algorithm, respectively.

Table 6.5. Confusion Matrix for the Schedule Detection Algorithm

		Identified Schedule	
		Occupied Schedule (C_0)	Unoccupied Schedule (C_1)
Actual Schedule	Occupied Schedule (C_0)	$N_{0,0}$ = Number of correctly identified C_0	$N_{0,1}$ = Number of C_0 incorrectly identified as C_1
	Unoccupied Schedule (C_1)	$N_{1,0}$ = Number of C_1 incorrectly identified as C_0	$N_{1,1}$ = Number of correctly identified C_1

Equation ((15) is used to estimate the accuracy of the schedule identification, which is the ratio of the number of correctly identified schedules ($N_{0,0}$+$N_{1,1}$) and total number of schedules ($N_{0,0}$+$N_{1,0}$+$N_{0,1}$+$N_{1,1}$). The goal of the schedule detection algorithm is to identify the occupied/unoccupied schedules with an accuracy of at least 85%.

$$Accuracy\ matrics\ (\%) = \left(\frac{(N_{0,0} + N_{1,1})}{(N_{0,0} + N_{1,0} + N_{0,1} + N_{1,1})}\right) \times 100 \quad (15)$$

Table 6.6 shows an example of the confusion matrix for the schedule detection algorithm (July 1 and July 2). The schedule detection algorithm identifies the schedules over the 24-hour period. The algorithm correctly identified the occupied schedule 8 times and incorrectly identified it 1 time. Also, the algorithm correctly identified the unoccupied schedule 14 times and incorrectly identified it 1 time. Therefore, the accuracy of schedule detection algorithm is 92%.

Table 6.6. Confusion Matrix for Occupancy Schedule Detection (Building A-RTU-3- July 1 and July 2)

Accuracy = 92%		Predicted Schedule	
		Occupied Schedule (C_0)	Unoccupied Schedule (C_1)
Actual Schedule	Occupied Schedule (C_0)	8	2
	Unoccupied Schedule (C_1)	0	14

Table 6.7 shows the example of the confusion matrix for the schedule detection algorithm (July 1 and July 6). The schedule detection algorithm identifies the schedules over the 24-hour period. The algorithm correctly identified the unoccupied schedule 24 times. Therefore, the accuracy of schedule detection algorithm is 100%.

Table 6.7. Confusion Matrix for Occupancy Schedule Detection (Building A-RTU-3- July 1 and July 6)

Accuracy =100%		Predicted Schedule	
		Occupied Schedule (C_0)	Unoccupied Schedule (C_1)
Actual Schedule	Occupied Schedule (C_0)	0	0
	Unoccupied Schedule (C_1)	0	24

6.5 Validation of Occupancy Schedule Detection Algorithm

As discussed in the previous section, a confusion matrix was used to quantify the accuracy of the schedule detection algorithm. Existing data were used to demonstrate and validate the performance of the scheduling detection algorithm. The occupancy scheduling was identified from the zone temperature time series and compared with supply fan signals (which is the ground truth). Figure 6.7 shows the example of the comparison between predicted and actual schedules, and the corresponding confusion matrix for HP-1 in Building C during the cooling season. The schedule detection algorithm estimated the occupancy schedules over the 24-hour period. The algorithm correctly predicted the occupied schedule for 11 hours and incorrectly predicted it for 1 hour. The algorithm also correctly predicted the unoccupied schedule for 10 hours and incorrectly predicted it for 2 hours. Therefore, the accuracy of schedule detection algorithm is 88% ([11+10]/24), satisfying the accuracy level of 85%.

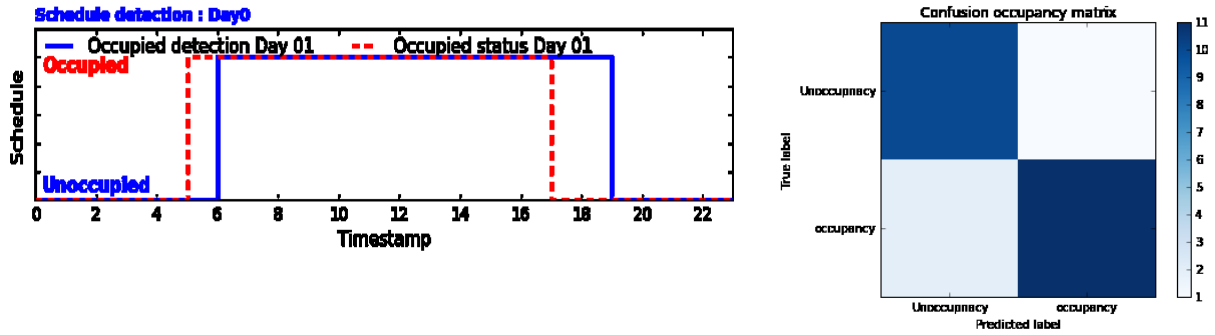

Figure 6.7. Comparison between Predicted and Actual Schedule and the Corresponding Confusion Matrix (Building C-HP-1-Summer-Day1)

Table 6.8 provides a summary of the occupancy schedule detection results for Building A. The results show that the algorithm detected the schedules correctly 50 times and incorrectly 60 times. All incorrect detections happened during summer and spring/fall seasons when the average outdoor temperatures was close to the zone set point during occupancy (between 69°F and 64°F). When the outdoor temperature is close to zone set point, RTUs generally start operating late morning or early afternoon; therefore, using just the zone temperature to detect the schedule failed because there is no variation in the zone temperature when the RTU is not actively cooling or heating.

Table 6.8. Summary of Occupancy Schedule Detection Results for Building F

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ ° F	Number of Correct Detection (≥85%)	Number of Incorrect Detection (< 85%)
B	Berkeley, CA	6	Summer	69	10	30
			Winter	51	28	0
			Spring/Fall	64	12	30
Total number of detections					50	60

Table 6.9 shows the summary of occupancy detection results for Building B. Unlike Building A, the results show higher detection accuracy—79 correct and only 2 incorrect detections. The average outdoor temperatures at this site were either lower than 50°F (during winter) or higher than 75°F (during summer).

Table 6.9. Summary of Occupancy Schedule Detection Results for Building B

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ ° F	Number of Correct Detection (≥85%)	Number of Incorrect Detection (< 85%)
A	Seattle, WA	4	Summer	77	27	1
			Winter	43	26	2
			Spring/Fall	53	26	2
The total number of detections					79	5

Table 6.10 summarizes the results of occupancy schedule detection for Building C and lists 82 correct and 64 incorrect detections. Most of the incorrect detections happened during winter and spring/fall when the average outdoor temperatures was between 50°F and 63°F. Although the average outdoor temperature was close to 50°F during winter, the number of incorrect detections was high because the building was open space and thus only a few of eight RTUs operated simultaneously.

Table 6.10. Summary of Occupancy Schedule Detection Results for Building C

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection (≥85%)	Number of Incorrect Detection (< 85%)
C	Richland, WA	8	Summer	73	48	1
			Winter	51	10	38
			Spring/Fall	63	24	25
Total number of detections					82	64

Table 6.11 summarizes the results for occupancy schedule detection for Building D and lists 11 correct and 18 incorrect detections. Fourteen out of 18 incorrect detections occurred during spring/fall when the average outdoor temperature was around 60°F.

Table 6.11. Summary of Occupancy Schedule Detection Results for Building D

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection (≥85%)	Number of Incorrect Detection (< 85%)
D	South Paris, ME	2	Summer	73	10	4
			Spring/Fall	60	1	14
Total number of detections					11	18

Table 6.12 summarizes the occupancy schedule detection results for Building E. The results also show a significantly high detection accuracy—23 correct and only 1 incorrect detection. The results were good even during the winter season.

Table 6.12. Summary of Occupancy Schedule Detection Results for Building E

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection (≥85%)	Number of Incorrect Detection (< 85%)
E	Miami, FL	2	Summer	84	14	0
			Winter	69	13	1
			Spring/Fall	80	14	0
Total number of detections					23	1

Table 6.13 shows the summary of occupancy schedule detection results for Building F. There were 23 correct and 19 incorrect detections for two RTUs.

Table 6.13. Summary of Occupancy Schedule Detection Results for Building F

Building	Location	Number of RTUs	Season	$T_{OA,AVG}$ °F	Number of Correct Detection (≥85%)	Number of Incorrect Detection (< 85%)
F	Cupertino, CA	2	Summer	73	7	7
			Winter	50	8	6
			Spring/Fall	61	8	6
Total number of detections					23	19

Table 6.14 shows the summary of schedule detection performance based on outdoor temperature range. The schedule detection algorithm detected occupied and unoccupied schedules correctly when outdoor temperatures were higher than 75°F (summer) and lower than 50°F (winter). The algorithm, however, showed some limitations when the outdoor air temperatures were between 50°F and 75°F (spring/fall). During this period, the systems are not typically running early in the day, making schedule detection difficult. The details of the results for each RTU and heat pump system in three buildings are presented in

Validation of Schedule Detection Algorithm. Overall, the occupancy schedule detection algorithm using the SAX method performs well under summer (hot) and winter (cold) weather conditions, when energy use is highest. The algorithm can also distinguish different schedules and generate clusters of occupancy scheduling based on zone temperature measurements alone.

Table 6.14. Summary of Schedule Detection Algorithm Results

Outdoor air temperature (°F)	Total Number of Correct Detected Schedules (>85%)	Total Number of Incorrect Detected Schedules (<85%)
Higher than 75°F	83	9
Between 50°F and 75°F	71	115
Lower than 50°F	128	46

7.0 How to Use the Algorithms to Improve SMBs Efficiency

Increased urgency to improve the operating efficiency of existing commercial building stock in the U.S. is driven by many reasons, chief among them being to mitigate the impacts of climate change. Many cities (e.g., New York, Seattle, etc.) are passing ordinances that require periodic retro-commissioning of commercial buildings. The U.S. climate action plan will give new impetus to many states and cities to follow the examples of Seattle and New York. Although traditional retro-commissioning processes can be effective, they cannot ensure the persistence of building operations beyond few months and they are also perceived to be costly. A technology-based solution can deliver retro-commissioning service at a lower cost and also ensure the persistence of building operations because it represents a continuous process.

The technology-based solution requires software algorithms and applications that can be deployed in the Cloud or on low-cost platforms (Raspberry PI and BeagleBone, etc.) in SMBs. The work reported in this report covers two such algorithms: 1) one to detect zone set points and 2) one to detect the number of RTU ON/OFF cycles. These algorithms can be used to ensure the persistence of building operations in SMBs by enforcing set points and mitigating the reasons for RTU short cycling. The algorithms can also automatically detect system overrides.

8.0 Conclusions

Three algorithms—to detect zone set points, RTU cycling, and occupancy schedule—were developed, tested, and validated.

To validate the algorithms, monitored data from number of zones/RTUs from six buildings in different climate locations were used. The set point and cycling detection algorithms use a peak detection technique that only requires one measured temperature to detect peaks/valleys in any given time-series data. The occupancy schedule detection algorithm was developed based on the symbolic mapping technique for time-series data. These algorithms do not require supervisory learning or additional sensor installation other than the zone temperature sensors.

The three different metrics were used to validate the three different algorithms. To evaluate the accuracy of algorithms, three different data sets for each zone/RTU were selected from the field data for different periods of the year (i.e., spring/fall, summer, and winter). Each selected data set contained five measurements (zone temperature, set point temperature, supply fan status, cooling/heating command, and outdoor temperature) of a 1-week period. Although only zone temperature was used by the algorithm, the remainder of the data was used as "ground truth" to verify the results of the algorithm.

Overall, the algorithms were successful in detecting the set points and ON/OFF cycles accurately using the peak detection technique. The occupancy schedule detection approach achieved an overall prediction accuracy of over 85% when outdoor temperatures were higher than 75°F (summer) and lower than 55°F (winter). Because these algorithms only use zone temperature for identification, if the RTUs are not actively heating/cooling, the algorithms will not be able to accurately identify set point, ON/OFF cycles, and schedules accurately.

9.0 References

ADM. 2009. *Market Assessment and Field M&V Study for Comprehensive Packaged A/C Systems Program, Final Report*. Sacramento, California.

Altman D, D Machin, T Bryant, and M Gardner (eds.) 2013. *Statistics with Confidence: Confidence Intervals and Statistical Guidelines.* John Wiley & Sons, Hoboken, New Jersey.

Ardehali MM and TF Smith. 2002. *Literature Review to Identify Existing Case Studies of Controls-Related Energy-Inefficiencies in Buildings*. ME-TFS-01-007, Department of Mechanical and Industrial Engineering, the University of Iowa, Iowa City, Iowa.

Azzini I, R Dell'Anna, F Ciocchetta, F Demichelis, A Sboner, E Blanzieri, A Malossini. 2004. Simple Methods for Peak Detection in Time Series Microarray Data. *In* Proceedings of CAMDA '04 (Critical Assessment of Microarray Data), Durham, North Carolina.

Breuker MS and JE Braun. 1998. Common Faults and their Impacts for Rooftop Air Conditioners. *HVAC&R Research* 4(3):303–318.

ISO (International Standards Organization). 1994. *Accuracy (Trueness and Precision) of Measurement Methods and Results – Part 1: General Principles and Definitions*. BS ISO 5725-1, Geneva, Switzerland.

Cormen TH. 2009. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts.

Coombes KR1, S Tsavachidis, JS Morris, KA Baggerly MC Hung and HM Kuerer. 2005. "Improved Peak Detection and Quantification of Mass Spectrometry Data Acquired from Surface-enhanced Laser Desorption and Ionization by Denoising Spectra with the Undecimated Discrete Wavelet Transform. *Proteomics* 5, 4107–4117.

Cowan A. 2004. *Review of Recent Commercial Rooftop Unit Field Studies in the Pacific Northwest and California*. Northwest Power and Conservation Council and Regional Technical Forum, Portland, Oregon.

D'Oca, S, S Corgnati, and T Hong. 2015. Data Mining of Occupant Behavior in Office Buildings. *Energy Procedia* 78:585–590.

EIA (Energy Information Administration). 2012. *Energy Information Administration Annual Energy Outlook 2012*. Available at: http://www.eia.gov/forecasts/aeo/.

Fukunaga K. 1990. *Introduction to Statistical Pattern Recognition.* Academic Press, West Lafayette, Indiana.

Goldin DQ and PC Kanellakis. 1995. On similarity queries for time-series data: constraint specification and implementation. *In* Proceedings of the International Conference on Principles and Practice of Constraint Programming (pp. 137–153). Springer: Berlin, Heidelberg.

Harmer K, G Howells, W Sheng, M Fairhurst, and F Deravi. 2008. A Peak-Trough Detection Algorithm Based on Momentum. *In* Proceedings of the IEEE Congress on Image and Signal Processing (CISP), pp. 454–458.

Jacobs P. 2003a. *Small HVAC Problems and Potential Savings Reports*. P500-03-082-A-25, California Energy Commission, Sacramento, California.

Jacobs P. 2003b. *Small HVAC System Design Guide*. Architectural Energy Corporation (AEC), Boulder, Colorado.

Jordanov VT, DL Hall, and M Kastner. 2002. Digital Peak Detector with Noise Threshold. *In* Proceedings of the IEEE Nuclear Science Symposium Conference 1:140–142.

Katipamula S and MR Brambley. 2004a. Fault Detection, Diagnostics, and Prognostics for Building Systems – A Review Part 2. *HVAC&R Research* 11(2):169–187.

Katipamula S and MR Brambley. 2004b. Fault Detection, Diagnostics, and Prognostics for Building Systems – A Review Part 1. *HVAC&R Research* 11(1):3–25.

Katipamula S, RM Underhill, JK Goddard, D Taasevigen, MA Piette, J Granderson, RE Brown, SM Lanzisera, and T Kuruganti. 2012. *Small-and Medium-Sized Commercial Building Monitoring and Controls Needs: A Scoping Stud*y. PPNL-22961, Pacific Northwest National Laboratory (PNNL), Richland, Washington.

Katipamula S, RG Lutes, G Hernandez, JN Haack, and BA Akyol. 2016a. "Transactional Network: Improving Efficiency and Enabling Grid Services for Building." Science and Technology for the Built Environment, 1-12. doi:10.1080/23744731.2016.1171628

Katipamula S, K Gowri, and G Hernandez. 2016b. An Open-source automated continuous condition-based maintenance platform for commercial buildings. *Science and Technology for the Built Environment* 00:1–10. doi: 10.1080/23744731.2016.1218236

Keogh E, K Chakrabarti, M Pazzani, and S Mehrotra. 2000. Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information System* 3(3):263–286.

Keogh, E. and Lin, J., 2005. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, *8*(2), pp.154-177.

Lin J, E Keogh, S Lonardi, and B Chiu. 2003. A Symbolic Representation of Time Series with Implications for Streaming Algorithms. *In* Proceedings of the 8th ACM (American Computing Machinery) SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2–11,Foster City, California.

Lin J, E Keogh, L Wei, and S Lonardi. 2007. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery* 15(2):107–144.

Mařík K, J Rojíček, P Stluka, and J Vass. 2011. Advanced HVAC control: Theory vs. reality. *In* IFAC (International Federation of Automatic Control) Proceedings. 44(1):3108–3113.

Mills E. 2011. Building commissioning: a golden opportunity for reducing energy costs and greenhouse gas emissions in the United States. *Energy Efficiency* 4(2):145–173.

Murphy J and N Maldeis. 2009. Using time-of-day scheduling to save energy. ASHRAE Journal, Vol.51 (5) pp. 42–49.

Orfanidis SJ. 1995. *Introduction to Signal Processing*. Prentice-Hall, Inc., Upper Saddle River, New Jersey.

Palshikar G. 2009. Simple algorithms for peak detection in time-series. Technical report'09. In Technical Report, TRDDC (Tata Research, Development and Design Center, Pune, India

Provost F and R Kohavi. 1998. Guest editors' introduction: On applied research in machine learning. *Machine Learning* 30(2):127–132.

Schneider R. 2011. Survey of Peaks/Valleys Identification in Time Series. University of Zurich, Department of Informatics, Switzerland.

Smith S. 2013. Digital signal processing: a practical guide for engineers and scientists. Newnes.

Stoppel CM and F Leite. 2014. Integrating probabilistic methods for describing occupant presence with building energy simulation models. *Energy and Buildings* 68:99–107.

Sun K, D Yana, T Hong, and S Guo. 2014. Stochastic modeling of overtime occupancy and its application in building energy simulation and calibration. *Building and Environment* 79:1–12.

Wang L, S Greenberg, J Fiegel, A Rubalcava, S Earni, X Pang, R Yin, S Woodworth, and J Hernandez-Maldonado. 2013. Monitoring-based HVAC commissioning of an existing office building for energy efficiency. *Applied Energy* 102:1382–1390.

Appendix A

Validation of Set Point Detection Algorithm

Appendix A

Validation of Set Point Detection Algorithm

Error! Reference source not found.shows the overall set point algorithm performance for 18 different RTU and heat pump systems at 3 different locations. The red, gray, and yellow shaded data indicate "incorrect detection," "unable to detect," and "at the accuracy threshold of ±1.0 °F," respectively.

Table A.1. Overall Set Point Detection Algorithm Performance

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
Building A	RTU1-Summer	Day1	74	74	73.7	-0.5	-0.2
		Day2	76	74	73.5	-0.7	-0.3
		Day3	73	74	73.6	-0.7	-0.2
		Day4	71	74	73.5	-0.8	-0.3
		Day5	76	74	73.6	-0.7	-0.1
		Day6	82	80	80.1	-1.0	1.3
		Day7	79	80	-	-	-
	RTU1-Winter	Day1	44	72	72.4	0.3	0.4
		Day2	41	72	72.3	0.2	0.3
		Day3	38	74	74.1	0.1	0.1
		Day4	38	74	74.1	0.0	0.1
		Day5	40	74	74.1	0.1	0.2
		Day6	48	80	-	-	-
		Day7	50	80	-	-	-
	RTU1-Spring/ Fall	Day1	57	75	75.4	0.3	0.4
		Day2	55	75	75.1	0.0	0.0
		Day3	53	72	72.2	-0.6	1.0
				74	73.4	-1.0	-0.3
		Day4	50	74	73.5	-0.5	-0.5
		Day5	51	74	73.5	0.0	0.0
		Day6	58	80	-	-	-
		Day7	58	80	-	-	-
	RTU2-Summer	Day1	76	74	74.3	0.1	0.3
				80	80.8	-1.3	2.8
		Day2	78	72	72.1	0.1	0.1
				80	80.6	0.5	0.8
		Day3	74	71	71.0	-0.3	0.2
		Day4	75	73	73.3	0.2	0.4
				80	80.1	-0.2	0.5
		Day5	78	73	73.1	-0.6	0.8
				80	80.6	0.5	0.6
		Day6	82	80	80.4	0.1	0.7
		Day7	77	80	80.4	0.1	0.6
	RTU2-Winter	Day1	42	75	75.1	-0.2	0.4
		Day2	39	80	-	-	-
		Day3	39	80	-	-	-
		Day4	35	75	75.1	-0.1	0.2
		Day5	37	75	75.1	-0.2	0.3
		Day6	45	80	-	-	-
		Day7	45	80	-	-	-
	RTU2-Spring/ Fall	Day1	57	73	73.5	0.4	0.6
		Day2	52	72	71.6	-0.8	0.0
		Day3	49	72	71.8	-0.3	-0.2
		Day4	47	72	71.8	-0.3	-0.2
		Day5	48	72	71.9	-0.3	0.0
		Day6	52	-	-	-	-
		Day7	54	-	-	-	-
	RTU3-Summer	Day1	75	72	72.1	0.1	0.2
		Day2	77	72	72.2	-0.3	0.5

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
		Day3	73	72	72.1	0.0	0.2
		Day4	73	72	72.8	-0.5	2.1
		Day5	77	72	72.0	-0.2	0.2
		Day6	84	80	80.8	0.5	1.1
		Day7	80	-	-	-	-
	RTU3-Winter	Day1	48	73	73.3	0.3	0.4
		Day2	44	73	73.3	0.3	0.4
		Day3	44	73	73.5	0.4	0.6
		Day4	44	73	73.4	0.3	0.5
		Day5	48	73	73.5	0.6	0.7
		Day6	61	80	-	-	-
		Day7	58	80	-	-	-
	RTU3-Spring/ Fall	Day1	59	73	73.5	0.4	0.6
		Day2	56	73	-	-	-
		Day3	54	73	72.4	-1.1	-0.1
		Day4	51	73	73.4	0.2	0.5
		Day5	52	73	73.3	0.2	0.5
		Day6	61	80	-	-	-
		Day7	60	80	-	-	-
	RTU4-Summer	Day1	75	74	74.3	-0.2	0.8
		Day2	78	74	74.3	0.0	0.7
		Day3	74	74	74.4	0.2	0.6
		Day4	73	74	74.6	0.1	1.1
		Day5	77	74	74.3	0.0	0.6
		Day6	83	80	80.8	-1.1	2.6
		Day7	77	80	-	-	-
	RTU4-Winter	Day1	44	72	71.8	-0.3	-0.2
		Day2	37	72	71.6	-0.6	-0.3
		Day3	29	72	71.6	-0.5	-0.3
				74	74.6	0.5	0.6
		Day4	33	74	74.2	0.0	0.3
		Day5	40	74	74.2	0.1	0.3
		Day6	47	80	-	-	-
		Day7	50	80	-	-	-
	RTU4-Spring/ Fall	Day1	54	72	72.4	0.4	0.5
		Day2	51	72	72.4	0.3	0.4
		Day3	49	72	72.3	0.3	0.3
		Day4	47	72	72.2	0.2	0.3
		Day5	43	72	72.2	0.2	0.3
		Day6	47	80	-	-	-
		Day7	49	80	-	-	-
Building B	RTU1-Summer	Day1	65	74	73.5	-0.6	-0.4
		Day2	65	74	73.5	-0.6	-0.4
		Day3	68	74	73.5	-0.7	-0.3
		Day4	71	74	73.5	-0.6	-0.4
		Day5	74	74	73.5	-0.8	-0.3
		Day6	80	80	80.6	0.6	0.7
		Day7	80	80	80.7	0.6	0.7
	RTU1-Winter	Day1	49	72	71.5	-0.6	-0.5
		Day2	49	72	71.6	-0.5	-0.3
		Day3	49	72	71.5	-0.6	-0.4
		Day4	51	72	71.7	-0.4	-0.3
		Day5	52	72	71.7	-0.3	-0.3
		Day6	56	-	-	-	-
		Day7	56	-	-	-	-
	RTU1-Spring/ Fall	Day1	56	72	71.8	-0.8	0.5
				76	76.2	0.0	0.5
		Day2	56	72	71.7	-0.4	-0.2
		Day3	62	72	71.5	2.4	-0.4
				76	75.7	-0.5	-0.2
		Day4	68	76	75.4	-0.7	-0.4
		Day5	64	76	75.7	-0.4	-0.3
		Day6	72	-	-	-	-
		Day7	69	-	-	-	-
		Day1	63	74	73.6	-0.8	0.0

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
	RTU2-Summer	Day2	64	74	73.7	-0.4	-0.3
		Day3	71	74	73.5	-0.6	-0.4
		Day4	75	74	73.4	-0.8	-0.3
		Day5	78	74	73.3	-0.8	-0.7
		Day6	86	-	-	-	-
		Day7	85	-	-	-	-
	RTU2-Winter	Day1	50	72	72.3	0.2	0.3
		Day2	48	72	72.1	-0.1	0.2
		Day3	50	72	71.9	-0.3	0.0
		Day4	49	72	71.9	-0.3	0.0
		Day5	48	72	71.9	-0.2	0.0
		Day6	55	-	-	-	-
		Day7	55	-	-	-	-
	RTU2-Spring/ Fall	Day1	61	72	72.5	0.4	0.5
		Day2	60	72	72.4	0.3	0.4
				76	75.5	-1.1	-0.3
		Day3	58	72	72.3	0.3	0.3
				76	75.8	-0.7	0.3
		Day4	61	72	72.0	-0.6	0.6
				76	76.6	0.3	0.9
		Day5	60	72	72.2	0.0	0.5
				76	75.5	-1.1	0.2
		Day6	66	-	-	-	-
		Day7	58	-	-	-	-
	RTU3-Summer	Day1	63	-	-	-	-
		Day2	65	71	71.5	0.1	0.9
		Day3	71	71	70.9	-0.5	0.3
				73	73.1	-0.4	0.6
		Day4	75	73	73.3	0.1	0.4
		Day5	77	73	73.2	0.2	0.3
		Day6	86	-	-	-	-
		Day7	86	-	-	-	-
	RTU3-Winter	Day1	49	72	72.0	-0.1	0.1
		Day2	49	72	72.1	0.1	0.2
		Day3	49	70	69.7	-0.7	0.0
				72	72.1	-0.5	0.3
		Day4	51	70	69.7	-0.3	-0.3
				72	71.9	-0.5	0.2
		Day5	52	72	71.9	-0.1	0.0
		Day6	56	-	-	-	-
		Day7	56	-	-	-	-
	RTU3-Spring/ Fall	Day1	56	70	69.2	-0.8	-0.7
		Day2	62	70	69.5	-0.6	-0.5
				74	73.3	-0.9	-0.6
		Day3	64	70	69.5	-0.5	-0.4
				74	73.1	-1.1	-0.7
		Day4	65	70	69.6	-0.5	-0.3
				74	73.3	-0.8	-0.7
		Day5	62	70	69.5	-0.6	-0.5
		Day6	66	-	-	-	-
		Day7	66	-	-	-	-
	RTU4-Summer	Day1	69	76	75.6	-0.4	-0.3
		Day2	63	-	-	-	-
		Day3	63	72	71.7	-0.5	-0.2
		Day4	60	72	71.5	-0.6	-0.5
				76	75.3	-0.8	-0.6
		Day5	63	72	71.5	-0.5	-0.5
				76	75.4	-0.9	-0.3
		Day6	74	-	-	-	-
		Day7	73	-	-	-	-
	RTU4-Winter	Day1	50	72	71.3	-0.8	-0.7
		Day2	48	72	71.1	-1.0	-0.8
		Day3	50	72	71.6	-0.6	-0.2
		Day4	49	72	71.3	-0.8	-0.5
		Day5	48	72	71.4	-0.9	-0.2

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
		Day6	55	60	59.5	-0.5	-1.5
		Day7	55	60	59.5	-0.5	-0.5
	RTU4-Spring/ Fall	Day1	61	72	72.3	0.3	0.4
		Day2	57	72	72.3	0.0	0.4
		Day3	60	72	72.4	0.2	0.5
		Day4	60	72	72.6	0.2	1.1
		Day5	61	72	72.5	0.5	0.6
		Day6	64	-	-	-	-
		Day7	61	-	-	-	-
	RTU5-Summer	Day1	73	76	75.3	-0.8	-0.7
		Day2	58	-	-	-	-
		Day3	60	-	-	-	-
		Day4	60	72	71.1	-1.8	-0.4
				76	75.5	-1.1	0.1
		Day5	65	72	71.3	-0.7	-0.7
				76	75.2	-0.8	-0.7
		Day6	77	-	-	-	-
	RTU5-Winter	Day1	62	-	-	-	-
		Day2	60	-	-	-	-
		Day3	61	-	-	-	-
		Day4	56	71	70.6	-0.4	-0.3
		Day5	53	-	-	-	-
		Day6	50	-	-	-	-
		Day7	54	-	-	-	-
	RTU5-Spring/ Fall	Day1	72	76	75.3	-0.8	-0.6
		Day2	66	76	75.2	-0.8	-0.7
		Day3	68	76	75.9	-2.1	1.9
		Day4	68	76	75.2	-1.0	-0.6
		Day5	57	76	74.9	-1.9	-0.4
		Day6	63	-	-	-	-
		Day7	77	-	-	-	-
	RTU6-Summer	Day1	73	74	73.5	-0.7	-0.4
		Day2	58	-	-	-	-
		Day3	60	-	-	-	-
		Day4	60	70	69.3	-0.8	-0.6
				74	73.4	-0.8	-0.3
		Day5	65	74	73.4	-0.7	-0.6
		Day6	77	-	-	-	-
	RTU6-Winter	Day1	58	-	-	-	-
		Day2	59	-	-	-	-
		Day3	62	-	-	-	-
		Day4	56	70	69.3	-0.7	-0.7
		Day5	53	-	-	-	-
	RTU6-Spring/ Fall	Day1	66	72	71.7	-0.4	-0.1
		Day2	65	72	71.6	-0.8	-0.1
		Day3	64	-	-	-	-
		Day4	66	-	-	-	-
		Day5	70	74	73.9	-0.3	0.1
		Day6	67	-	-	-	-
		Day7	73	-	-	-	-
Building C	HP1-Summer	Day1	88	74	74.5	0.3	0.6
				80	80.7	0.4	1.1
		Day2	88	72	72.9	0.7	1.0
		Day3	78	72	72.8	0.4	0.8
		Day4	73	72	-	-	-
		Day5	68	72	72.5	0.4	0.7
		Day6	68	80	80.6	0.5	0.7
		Day7	71	80	80.4	-0.2	0.9
	HP1-Winter	Day1	48	72	71.2	-1.2	-0.5
		Day2	46	72	71.5	-0.8	-0.1
		Day3	47	72	71.4	-1.0	-0.4
		Day4	46	72	71.4	-0.8	-0.4
		Day5	44	72	71.4	-0.8	-0.4
		Day6	49	80	-	-	-
		Day7	52	80	-	-	-

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
	HP1-Spring/ Fall	Day1	65	72	72.5	0.5	0.6
		Day2	65	72	72.9	0.7	1.2
		Day3	69	72	72.5	0.3	0.6
		Day4	73	72	72.8	0.5	1.0
		Day5	67	72	72.6	0.4	0.7
		Day6	65	80	-	-	-
		Day7	62	80	-	-	-
	HP2- Summer	Day1	88	72	72.7	0.4	0.9
		Day2	88	72	71.5	-1.7	0.7
		Day3	78	72	71.4	-0.9	-0.8
		Day4	73	72	71.7	0.6	0.7
		Day5	68	72	71.1	-1.2	0.5
		Day6	68	80	-	-	-
		Day7	71	80	-	-	-
	HP2-Winter	Day1	48	72	71.3	-0.9	-0.4
		Day2	46	72	71.5	-0.8	-0.1
		Day3	47	70	70.7	0.2	1.3
		Day4	46	70	69.7	-0.6	-0.1
				75	75.1	-0.5	0.7
		Day5	44	70	69.6	-0.9	0.0
				75	74.3	-1.8	0.3
		Day6	49	-	-	-	-
		Day7	52	70	70.8	0.6	1.0
	HP2-Spring/ Fall	Day1	65	72	72.6	0.3	0.9
		Day2	65	72	72.9	0.6	1.2
		Day3	69	72	72.8	0.3	1.9
		Day4	73	72	72.6	0.3	0.9
		Day5	67	72	72.8	0.5	1.0
		Day6	65	72	72.5	-0.2	1.3
		Day7	62	-	-	-	-
	HP3- Summer	Day1	89	72	74.9	2.6	3.1
		Day2	88	72	71.3	-0.5	1.6
		Day3	78	72	72.9	0.7	1.2
		Day4	73	72	72.7	0.6	0.7
		Day5	68	72	72.7	0.6	0.8
		Day6	68	72	-	-	-
		Day7	71	72	-	-	-
	HP3-Winter	Day1	48	72	71.4	-0.7	-0.6
		Day2	46	72	-	-	-
		Day3	47	72	-	-	-
		Day4	46	72	-	-	-
		Day5	44	72	71.4	-1.0	-0.2
		Day6	49	72	-	-	-
		Day7	52	72	-	-	-
	HP3-Spring/ Fall	Day1	65	74	-	-	-
		Day2	65	74	-	-	-
		Day3	69	74	74.5	0.3	0.8
		Day4	73	74	74.5	0.3	0.8
		Day5	67	74	74.6	0.4	0.7
		Day6	65	74	-	-	-
		Day7	62	74	-	-	-
	HP4- Summer	Day1	72	70	-	-	-
		Day2	64	70	-	-	-
		Day3	63	70	71.2	0.7	1.7
		Day4	67	70	71.1	0.7	1.5
		Day5	66	70	70.9	0.6	1.1
		Day6	64	70	-	-	-
		Day7	71	70	-	-	-
	HP4-Winter	Day1	48	72	-	-	-
		Day2	46	72	-	-	-
		Day3	47	72	-	-	-
		Day4	46	72	-	-	-
		Day5	44	72	-	-	-
		Day6	49	80	-	-	-
		Day7	52	80	-	-	-

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
	HP4-Spring/ Fall	Day1	57	71	71.4	0	0.5
		Day2	60	71	71.3	-0.1	0.6
		Day3	60	71	71.6	0.3	0.9
		Day4	64	71	71.4	0.3	0.5
		Day5	61	71	71.3	0.3	0.4
		Day6	63	70	-	-	-
		Day7	64	70	-	-	-
	HP5- Summer	Day1	88	72	72.9	0.7	1.0
		Day2	88	72	72.7	0.6	0.8
		Day3	78	72	72.6	0.5	0.7
		Day4	73	72	72.5	0.5	0.5
		Day5	68	72	72.6	0.4	0.8
		Day6	68	72	-	-	-
		Day7	71	72	-	-	-
	HP5-Winter	Day1	48	72	71.6	-0.5	-0.4
		Day2	46	72	71.6	-0.6	-0.3
		Day3	47	72	71.6	-0.5	-0.3
		Day4	46	72	71.5	-0.6	-0.5
		Day5	44	72	71.5	-0.6	-0.4
		Day6	49	72	-	-	-
		Day7	52	72	-	-	-
	HP5-Spring/ Fall	Day1	65	72	71.5	-0.7	-0.3
				76	76.3	-0.3	0.8
		Day2	65	72	71.6	-0.4	-0.3
				76	76.4	0.3	0.4
		Day3	69	72	71.7	-0.4	-0.3
				76	76.3	0.1	0.5
		Day4	73	76	76.3	0.1	0.9
		Day5	67	76	76.4	0.3	0.4
		Day6	65	76	-	-	-
		Day7	62	76	-	-	-
	HP6- Summer	Day1	88	74	74.7	-0.1	1.4
				80	80.3	0.5	0.7
		Day2	88	80	80.5	0.5	0.6
		Day3	78	72	72.6	0.5	0.7
		Day4	73	72	72.5	0.4	0.5
		Day5	68	72	72.6	0.5	0.6
		Day6	68	80	80.3	0.2	0.3
		Day7	71	80	80.3	0.2	0.3
	HP6-Winter	Day1	48	74	73.3	-0.7	-0.6
		Day2	46	74	73.3	-0.8	-0.7
		Day3	47	74	73.3	-0.8	-0.5
		Day4	46	74	73.4	-0.8	-0.5
		Day5	44	74	73.3	-0.7	-0.6
		Day6	49	74	-	-	-
		Day7	52	74	-	-	-
	HP6-Spring/ Fall	Day1	65	77	77.3	0.3	0.3
		Day2	65	77	77.5	0.4	0.6
				80	80.0	-0.5	0.6
		Day3	69	77	77.4	0.3	0.4
				80	80.3	0.3	0.4
		Day4	73	77	77.4	0.3	0.5
		Day5	67	77	77.3	0.1	0.4
		Day6	65	77	-	-	-
		Day7	62	77	-	-	-
	HP7- Summer	Day1	72	70	70.6	0.5	0.7
		Day2	64	70	70.6	0.6	0.7
		Day3	63	70	70.6	0.6	0.6
		Day4	67	70	70.5	0.5	0.6
		Day5	66	70	70.6	0.5	0.7
		Day6	64	70	-	-	-
		Day7	65	70	-	-	-
	HP7-Winter	Day1	48	69	68.6	-1.0	0.2
		Day2	46	69	69.0	-0.7	0.7
		Day3	47	69	68.9	-0.4	0.3

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
		Day4	46	69	69.7	-0.2	1.5
		Day5	44	69	69.2	-0.3	0.6
		Day6	49	69	-	-	-
		Day7	52	69	-	-	-
	HP7-Spring/ Fall	Day1	57	74	75.0	0.7	1.2
		Day2	61	74	74.4	0.0	0.8
		Day3	60	74	74.3	-0.5	1.1
		Day4	64	74	74.5	0.5	1.5
		Day5	61	74	74.0	-0.2	0.2
		Day6	63	74	-	-	-
		Day7	64	74	-	-	-
	HP8- Summer	Day1	88	74	74.5	0.3	0.6
		Day2	88	74	74.8	0.7	0.9
		Day3	78	74	74.6	0.5	0.7
		Day4	73	74	74.4	0.4	0.5
		Day5	68	74	74.5	0.4	0.7
		Day6	68	74	-	-	-
		Day7	71	74	-	-	-
	HP8-Winter	Day1	48	74	-	-	-
		Day2	46	74	-	-	-
		Day3	47	74	-	-	-
		Day4	46	74	-	-	-
		Day5	44	74	-	-	-
		Day6	49	74	-	-	-
		Day7	52	74	-	-	-
	HP8-Spring/ Fall	Day1	65	74	74.4	0.3	0.5
		Day2	65	74	74.4	0.3	0.4
		Day3	69	74	74.4	0.1	0.7
		Day4	73	74	74.4	0.2	0.6
		Day5	67	74	74.4	0.2	0.6
		Day6	65	74	-	-	-
		Day7	62	74	-	-	-
Building D	RTU-1- summer	Day1	68.2	71	71.4	71.2	71.7
		Day2	71.3	71	71.5	71.1	71.8
		Day3	75.4	71	71.5	71.5	71.6
		Day4	75.1	71	71.6	71.5	71.7
		Day5	75.4	71	71.6	71.6	71.6
		Day6	73.7	71	71.4	71.2	71.6
		Day7	68.1	71	71.5	71.4	71.5
	RTU-1- spring/fall	Day1	47.0	71	-	-	-
		Day2	58.4	71	-	-	-
		Day3	60.5	71	-	-	-
		Day4	59.2	71	-	-	-
		Day5	62.0	71	71.5	71.5	71.6
		Day6	64.7	71	71.6	71.5	71.6
		Day7	62.0	71	71.3	71.0	71.6
	RTU-2- summer	Day1	73.0	71	71.9	71.8	71.9
		Day2	67.8	71	71.7	71.6	71.7
		Day3	67.6	71	71.8	71.7	71.9
		Day4	71.6	71	71.6	71.6	71.7
		Day5	77.6	71	71.8	71.8	72.1
		Day6	70.7	71	71.7	71.6	71.7
		Day7	72.1	71	71.6	71.6	71.7
	RTU-2- spring/fall	Day1	47.7	71	-	-	-
		Day2	58.9	71	71.7	71.6	71.9
		Day3	61.3	71	71.7	71.5	71.8
		Day4	59.2	71	71.5	71.2	72.0
		Day5	62.5	71	71.7	71.7	71.8
		Day6	65.2	71	71.4	71.0	71.9
		Day7	62.0	71	71.5	71.4	71.6
Building E	RTU1- Summer	Day1	71.0	71.8	72.4	72.2	72.5
		Day2	70.8	71.8	72.4	72.0	72.8
		Day3	72.9	71.8	71.9	71.6	72.2
		Day4	72.9	71.8	72.0	71.6	72.5
		Day5	72.3	71.8	72.8	72.6	73.0

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
		Day6	79.2	71.8	72.3	72.1	72.5
		Day7	82.3	71.8	72.4	72.3	72.6
	RTU1-Winter	Day1	44.1	68.7	69.5	69.1	69.9
		Day2	44.2	68.7	70.6	70.5	70.8
		Day3	45.7	68.7	70.4	70.0	70.8
		Day4	46.8	70.7	-	-	-
		Day5	44.6	70.7	70.4	70.2	70.6
		Day6	47.3	70.7	71.0	70.7	71.2
		Day7	50.2	70.7	70.7	70.5	71.0
	RTU1-Spring/ Fall	Day1	56.2	70.7	71.7	71.5	71.9
		Day2	56.9	70.7	72.6	72.3	72.8
		Day3	61.8	74.7	72.9	72.7	73.1
		Day4	65.5	74.7	-	-	-
		Day5	60.1	74.7	-	-	-
		Day6	61.4	74.7	-	-	-
		Day7	61.0	74.7	74.1	73.7	74.5
	RTU2-Summer	Day1	78.7	72	72.5	72.5	72.6
		Day2	72.3	72	-	-	-
		Day3	71.7	72	72.5	72.5	72.6
		Day4	70.8	72	72.6	72.5	72.7
		Day5	70.7	72	72.5	72.5	72.5
		Day6	71.4	72	72.5	72.4	72.6
		Day7	70.5	72	72.4	72.1	72.8
	RTU2-Winter	Day1	51.6	71	69.8	69.5	70.2
		Day2	49.7	71	70.4	70.2	70.7
		Day3	51.3	71	70.2	70.0	70.4
		Day4	54.3	71	70.5	70.2	70.8
		Day5	54.7	71	70.6	70.3	70.8
		Day6	55.9	71	70.4	70.3	70.6
		Day7	63.0	71	70.8	70.6	71.1
	RTU2-Spring/ Fall	Day1	68.9	73	73.6	73.4	73.8
		Day2	68.0	73	73.6	73.3	73.8
		Day3	62.8	71	71.8	71.5	72.1
		Day4	56.7	71	70.7	70.4	71.0
		Day5	56.6	71	70.6	70.5	70.8
		Day6	55.5	71	70.4	70.0	70.7
		Day7	55.8	71	70.4	70.3	70.5
Building F	RTU1-Summer	Day1	83.1	70	70.9	70.5	71.3
		Day2	84.1	70	70.5	70.0	70.8
		Day3	84.4	70	70.1	69.8	70.4
		Day4	83.0	70	70.7	70.4	71.0
		Day5	82.5	70	70.8	70.7	70.9
		Day6	83.0	70	71.0	70.9	71.0
		Day7	86.3	70	70.7	70.3	71.0
	RTU1-Winter	Day1	64.2	70	70.3	70.0	70.6
		Day2	73.0	70	70.6	70.3	70.9
		Day3	75.3	70	70.6	70.5	70.7
		Day4	75.3	70	70.3	70.0	70.7
		Day5	63.3	70	70.5	70.4	70.7
		Day6	65.5	70	70.4	70.0	70.8
		Day7	71.0	70	70.7	70.6	70.8
	RTU1-Spring/ Fall	Day1	82.7	70	70.3	70.1	70.4
		Day2	80.1	70	70.5	70.3	70.7
		Day3	83.1	70	70.4	70.3	70.5
		Day4	82.0	70	70.3	70.1	70.6
		Day5	81.8	70	70.6	70.2	70.9
		Day6	82.4	70	70.5	70.3	70.6
		Day7	82.4	70	70.5	70.2	70.9
	RTU2-Summer	Day1	82.2	70	69.8	69.6	69.9
		Day2	77.0	70	69.2	69.1	69.3
		Day3	83.1	70	69.6	69.3	70.0
		Day4	85.0	70	70.4	70.4	70.5
		Day5	86.0	70	70.5	70.4	70.6
		Day6	86.9	70	70.6	70.5	70.8
		Day7	85.2	70	70.6	70.4	70.7

Building	Test code	Day	$T_{OD,avg}$ °F	T_{sp} °F	$T_{sp,pred}$ °F	T_{sp} with 90 [%] confidence interval °F	
	RTU2-Winter	Day1	64.1	70	70.0	69.9	70.2
		Day2	73.6	70	70.3	70.1	70.5
		Day3	75.3	70	69.3	69.2	69.4
		Day4	71.3	70	69.1	69.0	69.2
		Day5	62.8	70	69.7	69.3	70.0
		Day6	65.2	70	70.0	69.9	70.2
		Day7	70.3	70	70.4	70.2	70.5
	RTU2-Spring/ Fall	Day1	80.4	70	70.1	70.0	70.2
		Day2	77.8	70	70.3	70.2	70.3
		Day3	80.2	70	70.4	70.2	70.6
		Day4	80.7	70	70.2	70.1	70.3
		Day5	77.6	70	69.9	69.8	70.0
		Day6	79.0	70	69.8	69.5	70.0
		Day7	78.7	70	70.3	70.1	70.4

Appendix B

Validation of On/Off Cycle Algorithm

Appendix B

Validation of On/Off Cycle Algorithm

Table B.1 shows the overall ON-OFF cycle algorithm performance for 18 different RTU and heat pump systems at 3 different locations. The red, gray, and yellow shaded data indicate "incorrect detection," "unable to detect," and "at the accuracy threshold of 80 (%)," respectively.

Table B.1. Overall ON/OFF Cycle Detection Algorithm Performance

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
Building A	RTU1-Summer	Day1	74	7	6	14%	38	40	5%
		Day2	76	8	8	0%			
		Day3	73	5	6	20%			
		Day4	71	6	7	17%			
		Day5	76	6	6	0%			
		Day6	82	4	5	25%			
		Day7	79	2	2	0%			
	RTU1-Winter	Day1	44	14	17	21%	85	86	1%
		Day2	41	19	19	0%			
		Day3	38	18	18	0%			
		Day4	38	18	18	0%			
		Day5	40	16	14	13%			
		Day6	48	0	0	-			
		Day7	50	0	0	-			
	RTU1-Spring/ Fall	Day1	57	13	13	0%	65	64	2%
		Day2	55	11	11	0%			
		Day3	53	14	13	7%			
		Day4	50	14	14	0%			
		Day5	51	13	13	0%			
		Day6	58	0	0	-			
		Day7	58	0	0	-			
	RTU2-Summer	Day1	76	22	27	23%	235	233	1%
		Day2	78	26	27	4%			
		Day3	74	40	38	5%			
		Day4	75	37	39	5%			
		Day5	78	46	48	4%			
		Day6	82	47	34	28%			
		Day7	77	17	20	18%			
	RTU2-Winter	Day1	42	4	8	100%	20	34	70%
		Day2	39	3	3	0%			
		Day3	39	2	7	250%			
		Day4	35	2	6	200%			
		Day5	37	3	4	33%			
		Day6	45	2	3	50%			
		Day7	45	4	3	25%			
	RTU2-Spring/ Fall	Day1	57	15	16	7%	83	88	6%
		Day2	52	18	18	0%			
		Day3	49	15	16	7%			
		Day4	47	11	14	27%			
		Day5	48	19	19	0%			
		Day6	52	4	4	0%			
		Day7	54	1	1	0%			
	RTU3-Summer	Day1	75	7	7	0%	48	49	2%
		Day2	77	7	7	0%			
		Day3	73	6	9	50%			

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
		Day4	73	8	6	25%			
		Day5	77	8	10	25%			
		Day6	84	9	7	22%			
		Day7	80	3	3	0%			
	RTU3-Winter	Day1	48	13	12	8%	57	54	5%
		Day2	44	11	10	9%			
		Day3	44	10	9	10%			
		Day4	44	12	11	8%			
		Day5	48	10	9	10%			
		Day6	61	0	1	-			
		Day7	58	1	2	100%			
	RTU3-Spring/ Fall	Day1	59	8	7	13%	37	31	16%
		Day2	56	3	4	33%			
		Day3	54	7	5	29%			
		Day4	51	11	9	18%			
		Day5	52	8	6	25%			
		Day6	61	0	0	-			
		Day7	60	0	0	-			
	RTU4-Summer	Day1	75	13	17	31%	107	121	13%
		Day2	78	27	29	7%			
		Day3	74	16	19	19%			
		Day4	73	21	24	14%			
		Day5	77	25	25	0%			
		Day6	83	4	5	25%			
		Day7	77	1	2	100%			
	RTU4-Winter	Day1	44	44	44	0%	214	213	0%
		Day2	37	44	43	2%			
		Day3	29	45	45	0%			
		Day4	33	40	40	0%			
		Day5	40	41	41	0%			
		Day6	47	0	0	-			
		Day7	50	0	0	-			
	RTU4-Spring/ Fall	Day1	54	34	35	3%	204	204	0%
		Day2	51	40	39	3%			
		Day3	49	46	46	0%			
		Day4	47	44	43	2%			
		Day5	43	40	41	3%			
		Day6	47	0	0	-			
		Day7	49	0	0	-			
Building B	RTU1-Summer	Day1	65	6	7	17%	137	134	2%
		Day2	65	17	18	6%			
		Day3	68	28	27	4%			
		Day4	71	27	25	7%			
		Day5	74	33	31	6%			
		Day6	80	14	14	0%			
		Day7	80	12	12	0%			
	RTU1-Winter	Day1	49	40	41	3%	176	184	5%
		Day2	49	29	33	14%			
		Day3	49	21	26	24%			
		Day4	51	40	39	3%			
		Day5	52	46	45	2%			
		Day6	56	0	0	-			
		Day7	56	0	0	-			
	RTU1-Spring/ Fall	Day1	56	18	19	6%	130	116	11%
		Day2	56	27	27	0%			
		Day3	62	29	26	10%			
		Day4	68	29	21	28%			
		Day5	64	25	20	20%			
		Day6	72	2	2	0%			
		Day7	69	0	1	-			

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
	RTU2-Summer	Day1	63	2	6	200%	178	161	10%
		Day2	64	24	30	25%			
		Day3	71	49	36	27%			
		Day4	75	55	39	29%			
		Day5	78	46	46	0%			
		Day6	86	0	0	-			
		Day7	85	2	4	100%			
	RTU2-Winter	Day1	50	60	61	2%	283	287	1%
		Day2	48	53	56	6%			
		Day3	50	53	57	8%			
		Day4	49	57	55	4%			
		Day5	48	60	57	5%			
		Day6	55	0	1	-			
		Day7	55	0	0	-			
	RTU2-Spring/ Fall	Day1	61	15	19	27%	108	118	9%
		Day2	60	21	24	14%			
		Day3	58	23	27	17%			
		Day4	61	28	27	4%			
		Day5	60	21	21	0%			
		Day6	66	0	0	-			
		Day7	58	0	0	-			
	RTU3-Summer	Day1	63	0	3	-	89	99	11%
		Day2	65	12	15	25%			
		Day3	71	23	24	4%			
		Day4	75	26	25	4%			
		Day5	77	27	28	4%			
		Day6	86	0	1	-			
		Day7	86	1	3	200%			
	RTU3-Spring/ Fall	Day1	56	14	16	14%	61	72	18%
		Day2	62	12	16	33%			
		Day3	64	12	14	17%			
		Day4	65	12	14	17%			
		Day5	62	11	12	9%			
		Day6	66	0	0	-			
		Day7	66	0	0	-			
	RTU4-Summer	Day1	69	13	15	15%	52	65	25%
		Day2	63	0	4	-			
		Day3	63	5	7	40%			
		Day4	60	12	14	17%			
		Day5	63	15	18	20%			
		Day6	74	3	3	0%			
		Day7	73	4	4	0%			
	RTU4-Winter	Day1	50	27	27	0%	130	128	2%
		Day2	48	23	22	4%			
		Day3	50	27	25	7%			
		Day4	49	20	19	5%			
		Day5	48	14	16	14%			
		Day6	55	7	7	0%			
		Day7	55	12	12	0%			
	RTU4-Spring/ Fall	Day1	61	9	11	22%	75	79	5%
		Day2	57	24	23	4%			
		Day3	60	18	17	6%			
		Day4	60	11	15	36%			
		Day5	61	13	13	0%			
		Day6	64	0	0	-			
		Day7	61	0	0	-			
	RTU5-Summer	Day1	73	14	8	43%	51	41	20%
		Day2	58	1	1	0%			
		Day3	60	5	4	20%			
		Day4	60	15	11	27%			

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
		Day5	65	15	16	7%			
		Day6	77	1	1	0%			
	RTU5-Winter	Day1	62	0	0	-	32	27	16%
		Day2	60	1	2	100%			
		Day3	61	8	3	63%			
		Day4	56	22	21	5%			
		Day5	53	1	1	0%			
		Day6	50	0	0	-			
		Day7	54	0	0	-			
	RTU5-Spring/ Fall	Day1	72	12	9	25%	60	48	20%
		Day2	66	15	11	27%			
		Day3	68	13	10	23%			
		Day4	68	12	10	17%			
		Day5	57	7	7	0%			
		Day6	63	0	0	-			
		Day7	77	1	1	0%			
	RTU6-Summer	Day1	73	15	12	20%	42	41	2%
		Day2	58	1	2	100%			
		Day3	60	1	2	100%			
		Day4	60	10	11	10%			
		Day5	65	15	14	7%			
		Day6	77	0	0	-			
	RTU6-Winter	Day1	58	1	0	100%	17	19	12%
		Day2	59	0	0	-			
		Day3	62	4	4	-			
		Day4	56	12	15	25%			
		Day5	53	0	0	0%			
	RTU6-Spring/ Fall	Day1	66	13	10	23%	42	34	19%
		Day2	65	12	9	25%			
		Day3	64	6	5	17%			
		Day4	66	3	4	33%			
		Day5	70	8	6	25%			
		Day6	67	0	0	-			
		Day7	73	0	0	-			
Building C	HP1-Summer	Day1	88	22	23	5%	92	95	3%
		Day2	88	25	21	16%			
		Day3	78	14	16	14%			
		Day4	73	4	5	25%			
		Day5	68	12	11	8%			
		Day6	68	5	6	20%			
		Day7	71	10	13	30%			
	HP1-Winter	Day1	48	13	13	0%	43	40	7%
		Day2	46	8	8	0%			
		Day3	47	7	6	14%			
		Day4	46	7	6	14%			
		Day5	44	8	7	13%			
		Day6	49	0	0	0%			
		Day7	52	0	0	0%			
	HP1-Spring/ Fall	Day1	65	15	14	7%	90	84	7%
		Day2	65	12	13	8%			
		Day3	69	18	16	11%			
		Day4	73	24	21	13%			
		Day5	67	21	20	5%			
		Day6	65	0	0	0%			
		Day7	62	0	0	0%			
	HP2-Summer	Day1	88	8	5	38%	87	93	7%
		Day2	88	3	5	67%			
		Day3	78	12	14	17%			
		Day4	73	16	18	13%			

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
		Day5	68	16	15	6%			
		Day6	68	12	14	17%			
		Day7	71	20	22	10%			
	HP2-Winter	Day1	48	23	21	9%	102	102	0%
		Day2	46	13	16	23%			
		Day3	47	14	14	0%			
		Day4	46	12	13	8%			
		Day5	44	15	17	13%			
		Day6	49	13	10	23%			
		Day7	52	12	11	8%			
	HP2-Spring/ Fall	Day1	65	21	21	0%	98	87	11%
		Day2	65	12	12	0%			
		Day3	69	13	13	0%			
		Day4	73	24	16	33%			
		Day5	67	15	11	27%			
		Day6	65	7	10	43%			
		Day7	62	6	4	33%			
	HP3-Summer	Day1	89	7	15	114%	16	78	388%
		Day2	88	2	17	750%			
		Day3	78	2	10	400%			
		Day4	73	2	15	650%			
		Day5	68	1	12	1100%			
		Day6	68	0	6	-			
		Day7	71	2	3	50%			
	HP3-Winter	Day1	48	10	16	60%	36	39	8%
		Day2	46	4	4	0%			
		Day3	47	5	6	20%			
		Day4	46	8	5	38%			
		Day5	44	9	8	11%			
		Day6	49	0	0	0%			
		Day7	52	0	0	0%			
	HP3-Spring/ Fall	Day1	65	1	1	0%	6	28	367%
		Day2	65	1	2	100%			
		Day3	69	1	7	600%			
		Day4	73	2	10	400%			
		Day5	67	1	8	700%			
		Day6	65	0	0	-			
		Day7	62	0	0	-			
	HP4-Summer	Day1	72	3	5	67%	23	27	17%
		Day2	64	7	6	14%			
		Day3	63	6	6	0%			
		Day4	67	5	5	0%			
		Day5	66	2	5	150%			
		Day6	64	0	0	-			
		Day7	71	0	0	-			
	HP4-Winter	Day1	48	3	4	33%	5	17	240%
		Day2	46	2	4	100%			
		Day3	47	0	2	-			
		Day4	46	0	0	-			
		Day5	44	0	3	-			
		Day6	49	0	2	-			
		Day7	52	0	2	-			
	HP4-Spring/ Fall	Day1	57	12	9	25%	56	55	2%
		Day2	60	10	6	40%			
		Day3	60	14	12	14%			
		Day4	64	8	9	13%			
		Day5	61	12	13	8%			
		Day6	63	0	0	0%			
		Day7	64	0	6	0%			

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
	HP5-Summer	Day1	88	28	23	18%	153	133	13%
		Day2	88	39	33	15%			
		Day3	78	17	19	12%			
		Day4	73	36	23	36%			
		Day5	68	29	28	3%			
		Day6	68	0	1	0%			
		Day7	71	4	6	50%			
	HP5-Winter	Day1	48	27	30	11%	116	109	6%
		Day2	46	28	26	7%			
		Day3	47	13	11	15%			
		Day4	46	26	24	8%			
		Day5	44	22	18	18%			
		Day6	49	0	0	0%			
		Day7	52	0	0	0%			
	HP5-Spring/ Fall	Day1	65	31	32	3%	158	173	9%
		Day2	65	28	33	18%			
		Day3	69	29	33	14%			
		Day4	73	36	38	6%			
		Day5	67	34	37	9%			
		Day6	65	0	0	0%			
		Day7	62	0	0	0%			
	HP6-Summer	Day1	88	29	28	3%	179	168	6%
		Day2	88	25	23	8%			
		Day3	78	10	9	10%			
		Day4	73	25	20	20%			
		Day5	68	28	28	0%			
		Day6	68	27	21	22%			
		Day7	71	35	39	11%			
	HP6-Winter	Day1	48	42	41	2%	150	140	7%
		Day2	46	31	28	10%			
		Day3	47	18	16	11%			
		Day4	46	30	32	7%			
		Day5	44	29	23	21%			
		Day6	49	0	0	-			
		Day7	52	0	0	-			
	HP6-Spring/ Fall	Day1	65	29	30	3%	164	175	7%
		Day2	65	28	30	7%			
		Day3	69	36	39	8%			
		Day4	73	39	41	5%			
		Day5	67	32	35	9%			
		Day6	65	0	0	0%			
		Day7	62	0	0	0%			
	HP7-Summer	Day1	72	29	22	24%	135	112	17%
		Day2	64	26	21	19%			
		Day3	63	24	21	13%			
		Day4	67	32	23	28%			
		Day5	66	16	17	6%			
		Day6	64	3	2	33%			
		Day7	65	5	6	20%			
	HP7-Winter	Day1	48	9	13	44%	18	34	89%
		Day2	46	4	5	25%			
		Day3	47	3	5	67%			
		Day4	46	1	3	200%			
		Day5	44	1	8	700%			
		Day6	49	0	0	-			
		Day7	52	0	0	-			
	HP7-Spring/ Fall	Day1	57	15	8	47%	68	57	16%
		Day2	61	17	10	41%			
		Day3	60	8	9	13%			
		Day4	64	22	15	32%			

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
		Day5	61	1	7	600%			
		Day6	63	2	5	150%			
		Day7	64	3	3	0%			
	HP8-Summer	Day1	88	27	23	15%	100	87	13%
		Day2	88	22	19	14%			
		Day3	78	13	12	8%			
		Day4	73	24	21	13%			
		Day5	68	14	12	14%			
		Day6	68	0	0	-			
		Day7	71	0	0	-			
	HP8-Winter	Day1	48	2	0	100%	9	14	56%
		Day2	46	2	2	0%			
		Day3	47	3	2	33%			
		Day4	46	0	4	-			
		Day5	44	2	3	50%			
		Day6	49	0	2	-			
		Day7	52	0	1	-			
	HP8-Spring/ Fall	Day1	65	16	14	13%	112	96	14%
		Day2	65	22	18	18%			
		Day3	69	25	23	8%			
		Day4	73	26	22	15%			
		Day5	67	23	19	17%			
		Day6	65	0	0	-			
		Day7	62	0	0	-			
Building D	RTU-1 summer	Day1	68.2	15	17	87%	197	199	99%
		Day2	71.3	23	22	96%			
		Day3	75.4	34	32	94%			
		Day4	75.1	35	36	97%			
		Day5	75.4	37	37	100%			
		Day6	73.7	21	23	90%			
		Day7	68.1	32	32	100%			
	RTU-1 spring/fall	Day1	47.0	0	-	-	26	26	100%
		Day2	58.4	0	-	-			
		Day3	60.5	1	-	-			
		Day4	59.2	3	-	-			
		Day5	62.0	10	11	90%			
		Day6	64.7	9	10	89%			
		Day7	62.0	3	5	60%			
	RTU-2 summer	Day1	73.0	13	14	92%	138	148	93%
		Day2	67.8	32	31	97%			
		Day3	67.6	16	19	84%			
		Day4	71.6	21	23	90%			
		Day5	77.6	13	14	92%			
		Day6	70.7	27	27	100%			
		Day7	72.1	16	20	80%			
	RTU-2 spring/fall	Day1	47.7	0	-	-	65	75	85%
		Day2	58.9	10	13	77%			
		Day3	61.3	10	9	90%			
		Day4	59.2	10	11	90%			
		Day5	62.5	13	17	77%			
		Day6	65.2	13	16	82%			
		Day7	62.0	9	9	100%			
Building E	RTU-1 summer	Day1	71.0	11	9	82%	60	63	95%
		Day2	70.8	9	8	89%			
		Day3	72.9	7	15	47%			
		Day4	72.9	9	6	67%			
		Day5	72.3	7	9	78%			
		Day6	79.2	6	6	100%			
		Day7	82.3	11	10	91%			

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
	RTU-1 winter	Day1	44.1	1	-	-	26	56	46%
		Day2	44.2	1	8	13%			
		Day3	45.7	3	8	38%			
		Day4	46.8	2	-	-			
		Day5	44.6	5	7	71%			
		Day6	47.3	7	17	41%			
		Day7	50.2	7	12	58%			
	RTU-1 spring/fall	Day1	56.2	1	6	17%	18	40	45%
		Day2	56.9	1	8	13%			
		Day3	61.8	3	6	50%			
		Day4	65.5	5	-	-			
		Day5	60.1	1	-	-			
		Day6	61.4	5	-	-			
		Day7	61.0	2	6	33%			
	RTU-2 summer	Day1	78.7	13	10	77%	85	74	87%
		Day2	72.3	5	-	-			
		Day3	71.7	17	13	76%			
		Day4	70.8	5	6	83%			
		Day5	70.7	14	12	86%			
		Day6	71.4	7	11	64%			
		Day7	70.5	24	22	92%			
	RTU-2 winter	Day1	51.6	7	7	100%	63	77	82%
		Day2	49.7	17	22	77%			
		Day3	51.3	7	11	64%			
		Day4	54.3	13	16	81%			
		Day5	54.7	7	9	78%			
		Day6	55.9	6	7	86%			
		Day7	63.0	6	5	83%			
	RTU-2 spring/fall	Day1	68.9	9	10	90%	43	58	74%
		Day2	68.0	8	11	73%			
		Day3	62.8	2	11	18%			
		Day4	56.7	7	6	86%			
		Day5	56.6	5	8	63%			
		Day6	55.5	5	6	83%			
		Day7	55.8	7	6	86%			
Building F	RTU-1 summer	Day1	83.1	39	29	74%	253	225	89%
		Day2	84.1	43	34	80%			
		Day3	84.4	18	25	72%			
		Day4	83.0	55	35	64%			
		Day5	82.5	46	38	83%			
		Day6	83.0	32	33	97%			
		Day7	86.3	20	31	65%			
	RTU-1 winter	Day1	64.2	32	30	94%	174	177	98%
		Day2	73.0	26	25	96%			
		Day3	75.3	12	18	67%			
		Day4	75.3	12	28	43%			
		Day5	63.3	28	24	86%			
		Day6	65.5	36	30	83%			
		Day7	71.0	28	22	80%			
	RTU-1 spring/fall	Day1	82.7	35	31	89%	240	231	96%
		Day2	80.1	39	34	87%			
		Day3	83.1	39	35	90%			
		Day4	82.0	35	32	91%			
		Day5	81.8	38	30	80%			
		Day6	82.4	30	38	80%			
		Day7	82.4	24	31	77%			
	RTU-2 summer	Day1	82.2	7	37	19%	139	217	44%
		Day2	77.0	12	29	41%			
		Day3	83.1	17	36	47%			

Building	Test code	Day	$T_{OD,avg}$ °F	ON/OFF cycling [ea]	ON/OFF cycling detection [ea]	MAPE [%]	Weekly ON/OFF cycling [ea]	Weekly ON/OFF cycling detection [ea]	Weekly MAPE [%]
		Day4	85.0	23	38	61%			
		Day5	86.0	24	25	96%			
		Day6	86.9	32	27	84%			
		Day7	85.2	24	25	96%			
	RTU-2 winter	Day1	64.1	51	33	65%	248	220	89%
		Day2	73.6	28	34	82%			
		Day3	75.3	16	30	53%			
		Day4	71.3	38	23	61%			
		Day5	62.8	47	35	74%			
		Day6	65.2	40	35	88%			
		Day7	70.3	28	30	93%			
	RTU-2 spring/fall	Day1	80.4	8	30	27%	53	183	-145%
		Day2	77.8	10	26	38%			
		Day3	80.2	8	23	35%			
		Day4	80.7	5	24	21%			
		Day5	77.6	8	27	30%			
		Day6	79.0	6	26	23%			
		Day7	78.7	8	27	30%			

Appendix C

Validation of Schedule Detection Algorithm

Appendix C

Validation of Schedule Detection Algorithm

Table C.1 shows the overall schedule algorithm performance for 18 different RTU and heat pump systems at 3 different locations. The red, gray, and yellow shaded data indicate "incorrect detection," "unable to detect," and "at the accuracy threshold of 85 (%)," respectively.

Table C.1. Overall Schedule Detection Algorithm Performance

Building	Test code	Day	$T_{OD, avg}$ °F	Unoccpancy/ $Unoccpancy_{pred}$	Unoccpancy/ $Occpancy_{pred}$	Occpancy/ $Unoccpancy_{pred}$	Occpancy/ $Occpancy_{pred}$	Accuracy	Weekly Accuracy
Building A	RTU1-Summer	Day1	74	11	0	2	11	92%	93%
		Day2	76	11	0	2	11	92%	
		Day3	73	11	0	2	11	92%	
		Day4	71	11	0	2	11	92%	
		Day5	76	9	0	3	11	88%	
		Day6	82	24	0	0	0	100%	
		Day7	79	24	0	0	0	100%	
	RTU1-Winter	Day1	44	10	0	3	10	88%	90%
		Day2	41	10	1	2	11	88%	
		Day3	38	10	1	2	11	88%	
		Day4	38	10	1	2	11	88%	
		Day5	40	10	1	2	11	88%	
		Day6	48	24	0	0	0	100%	
		Day7	50	24	0	0	0	100%	
	RTU1-Spring/ Fall	Day1	57	10	1	2	11	88%	93%
		Day2	55	12	0	1	11	96%	
		Day3	53	11	1	2	10	88%	
		Day4	50	11	0	2	11	92%	
		Day5	51	11	0	2	11	92%	
		Day6	58	24	0	0	0	100%	
		Day7	58	24	0	0	0	100%	
	RTU2-Summer	Day1	76	5	6	4	9	58%	89%
		Day2	78	11	2	1	10	88%	
		Day3	74	11	0	0	13	100%	
		Day4	75	12	0	1	11	96%	
		Day5	78	12	0	2	10	92%	
		Day6	82	21	3	0	0	88%	
		Day7	77	24	0	0	0	100%	
	RTU2-Winter	Day1	42	10	1	2	11	88%	86%
		Day2	39	10	1	2	11	88%	
		Day3	39	9	3	3	9	75%	
		Day4	35	10	2	1	11	88%	
		Day5	37	10	2	1	11	88%	
		Day6	45	24	0	0	0	100%	
		Day7	45	18	0	5	1	79%	
	RTU2-Spring/ Fall	Day1	57	10	0	3	11	88%	92%
		Day2	52	11	0	2	11	92%	
		Day3	49	11	0	2	11	92%	
		Day4	47	11	0	2	11	92%	
		Day5	48	10	0	3	11	88%	
		Day6	52	22	0	2	0	92%	
		Day7	54	24	0	0	0	100%	
	RTU3-Summer	Day1	75	12	0	1	11	96%	95%
		Day2	77	10	0	3	11	88%	
		Day3	73	11	0	3	11	92%	
		Day4	73	12	0	1	11	96%	
		Day5	77	11	1	1	11	92%	
		Day6	84	24	0	0	0	100%	
		Day7	80	24	0	0	0	100%	

Building	Test code	Day	$T_{OD, avg}$ °F	Unoccpancy/ $Unoccpancy_{pred}$	Unoccpancy/ $Occpancy_{pred}$	Occpancy/ $Unoccpancy_{pred}$	Occpancy/ $Occpancy_{pred}$	Accuracy	Weekly Accuracy
	RTU3-Winter	Day1	48	11	0	1	12	96%	
		Day2	44	10	0	3	11	88%	
		Day3	44	10	1	3	11	88%	
		Day4	44	10	1	2	11	88%	90%
		Day5	48	11	1	1	11	92%	
		Day6	61	10	1	2	11	88%	
		Day7	58	11	1	1	11	92%	
	RTU3-Spring/ Fall	Day1	59	9	0	4	11	83%	
		Day2	56	10	0	3	11	88%	
		Day3	54	10	0	3	11	88%	
		Day4	51	9	0	4	11	83%	90%
		Day5	52	10	1	2	11	88%	
		Day6	61	24	0	0	0	100%	
		Day7	60	24	0	0	0	100%	
	RTU4-Summer	Day1	75	12	0	1	11	96%	
		Day2	78	11	0	3	10	88%	
		Day3	74	11	0	3	10	88%	
		Day4	73	10	0	3	11	88%	89%
		Day5	77	10	0	3	11	88%	
		Day6	83	10	0	3	11	88%	
		Day7	77	10	0	3	11	88%	
	RTU4-Winter	Day1	44	16	1	1	6	92%	
		Day2	37	14	0	1	9	96%	
		Day3	29	12	0	2	10	92%	
		Day4	33	11	0	2	11	92%	92%
		Day5	40	11	0	2	11	92%	
		Day6	47	11	0	2	11	92%	
		Day7	50	11	0	2	11	92%	
	RTU4-Spring/ Fall	Day1	54	11	0	2	11	92%	
		Day2	51	10	0	3	11	88%	
		Day3	49	11	0	1	12	96%	
		Day4	47	11	0	1	12	96%	94%
		Day5	43	11	0	1	12	96%	
		Day6	47	11	0	1	12	96%	
		Day7	49	11	0	1	12	96%	
Building B	RTU1-Summer	Day1	65	6	0	5	13	79%	
		Day2	65	7	0	5	12	79%	
		Day3	68	2	0	9	13	63%	
		Day4	71	6	0	7	11	71%	80%
		Day5	74	10	0	2	12	92%	
		Day6	80	21	0	3	0	79%	
		Day7	80	21	0	3	0	79%	
	RTU1-Winter	Day1	49	10	0	2	12	92%	
		Day2	49	14	0	1	9	96%	
		Day3	49	10	0	3	11	88%	
		Day4	51	10	0	2	12	92%	93%
		Day5	52	8	1	2	13	88%	
		Day6	56	24	0	0	0	100%	
		Day7	56	24	0	0	0	100%	
	RTU1-Spring/ Fall	Day1	56	9	3	5	5	58%	
		Day2	56	9	3	7	7	67%	
		Day3	62	7	5	6	6	54%	
		Day4	68	8	5	6	5	54%	60%
		Day5	64	10	3	8	3	54%	
		Day6	72	14	0	10	0	58%	
		Day7	69	18	0	6	0	75%	
	RTU2-Summer	Day1	63	10	0	7	7	71%	
		Day2	64	9	0	8	7	67%	
		Day3	71	7	0	5	12	79%	
		Day4	75	12	2	0	10	92%	85%
		Day5	78	13	4	0	7	83%	
		Day6	86	24	0	0	0	100%	
		Day7	85	24	0	0	0	100%	
		Day1	50	10	0	2	12	92%	95%

Building	Test code	Day	$T_{OD, avg}$ °F	Unoccpancy/ $Unoccpancy_{pred}$	Unoccpancy/ $Occpancy_{pred}$	Occpancy/ $Unoccpancy_{pred}$	Occpancy/ $Occpancy_{pred}$	Accuracy	Weekly Accuracy
	RTU2-Winter	Day2	48	11	0	1	12	96%	
		Day3	50	11	0	1	12	96%	
		Day4	49	10	0	2	12	92%	
		Day5	48	10	0	2	12	92%	
		Day6	55	24	0	0	0	100%	
		Day7	55	24	0	0	0	100%	
	RTU2-Spring/Fall	Day1	61	6	6	6	6	50%	67%
		Day2	60	7	7	5	5	50%	
		Day3	58	8	4	6	6	58%	
		Day4	61	11	2	8	3	58%	
		Day5	60	6	6	6	6	50%	
		Day6	66	24	0	0	0	100%	
		Day7	58	24	0	0	0	100%	
	RTU3-Summer	Day1	63	15	7	2	0	63%	74%
		Day2	65	5	0	8	11	67%	
		Day3	71	12	6	1	5	71%	
		Day4	75	12	7	1	4	67%	
		Day5	77	13	11	0	0	54%	
		Day6	86	24	0	0	0	100%	
		Day7	86	24	0	0	0	100%	
	RTU3-Winter	Day1	49	11	0	2	11	92%	92%
		Day2	49	11	0	2	11	92%	
		Day3	49	11	0	2	11	92%	
		Day4	51	9	0	3	12	88%	
		Day5	52	7	1	2	13	88%	
		Day6	56	24	0	0	0	100%	
		Day7	56	24	0	0	0	100%	
	RTU3-Spring/Fall	Day1	56	10	12	0	2	50%	64%
		Day2	62	8	3	7	6	58%	
		Day3	64	8	4	7	5	54%	
		Day4	65	9	5	7	3	50%	
		Day5	62	9	4	7	4	54%	
		Day6	66	22	2	0	0	92%	
		Day7	66	22	0	0	0	92%	
	RTU4-Summer	Day1	69	13	9	0	2	63%	70%
		Day2	63	9	3	4	8	71%	
		Day3	63	10	4	3	7	71%	
		Day4	60	9	4	3	8	71%	
		Day5	63	11	5	1	7	75%	
		Day6	74	13	0	11	0	54%	
		Day7	73	20	0	4	0	83%	
	RTU4-Winter	Day1	50	10	1	1	12	92%	93%
		Day2	48	10	1	1	12	92%	
		Day3	50	8	1	2	13	88%	
		Day4	49	11	1	1	11	92%	
		Day5	48	10	1	1	12	92%	
		Day6	55	24	0	0	0	100%	
		Day7	55	24	0	0	0	100%	
	RTU4-Spring/Fall	Day1	61	4	0	3	17	88%	93%
		Day2	57	11	0	2	11	92%	
		Day3	60	10	0	2	12	92%	
		Day4	60	10	0	3	11	88%	
		Day5	61	10	0	2	12	92%	
		Day6	64	24	0	0	0	100%	
		Day7	61	24	0	0	0	100%	
	RTU5-Summer	Day1	73	16	3	4	1	71%	49%
		Day2	58	3	8	1	12	63%	
		Day3	60	0	0	11	13	54%	
		Day4	60	3	8	4	9	50%	
		Day5	65	8	6	5	5	54%	
		Day6	77	13	11	0	0	54%	
	RTU5-Spring/Fall	Day1	72	1	9	2	12	54%	61%
		Day2	66	3	7	4	10	54%	
		Day3	68	6	7	1	10	67%	

Building	Test code	Day	$T_{OD, avg}$ °F	Unoccpancy/ $Unoccpancy_{pred}$	Unoccpancy/ $Occpancy_{pred}$	Occpancy/ $Unoccpancy_{pred}$	Occpancy/ $Occpancy_{pred}$	Accuracy	Weekly Accuracy
		Day4	68	1	8	4	11	50%	
		Day5	57	3	9	1	11	58%	
		Day6	63	9	0	7	8	71%	
		Day7	77	16	0	6	2	75%	
	RTU6-Summer	Day1	73	0	3	0	20	88%	58%
		Day2	58	5	7	0	12	71%	
		Day3	60	4	5	5	10	58%	
		Day4	60	4	7	3	10	58%	
		Day5	65	5	3	5	8	54%	
		Day6	77	20	0	3	0	88%	
	RTU6-Spring/ Fall	Day1	66	6	0	4	14	83%	71%
		Day2	65	8	0	2	14	92%	
		Day3	64	5	9	2	8	54%	
		Day4	66	1	9	1	13	58%	
		Day5	70	0	0	10	14	58%	
		Day6	67	16	0	8	0	67%	
		Day7	73	20	0	4	0	83%	
Building C	HP1-Summer	Day1	88	15	0	0	9	100%	95%
		Day2	88	6	3	0	15	88%	
		Day3	78	15	0	1	8	96%	
		Day4	73	12	0	1	11	96%	
		Day5	68	11	0	3	10	88%	
		Day6	68	14	0	0	10	100%	
		Day7	71	23	0	1	0	96%	
	HP1-Winter	Day1	48	9	1	5	9	75%	70%
		Day2	46	8	3	5	8	67%	
		Day3	47	5	2	8	9	58%	
		Day4	46	7	0	8	9	67%	
		Day5	44	6	1	8	9	63%	
		Day6	49	17	0	7	0	71%	
		Day7	52	21	0	3	0	88%	
	HP1-Spring/ Fall	Day1	65	9	0	6	9	75%	74%
		Day2	65	11	0	4	9	83%	
		Day3	69	10	0	5	9	79%	
		Day4	73	13	0	2	9	92%	
		Day5	67	8	0	7	9	71%	
		Day6	65	14	0	9	0	58%	
		Day7	62	15	0	9	0	63%	
	HP2-Summer	Day1	88	14	3	0	7	88%	88%
		Day2	88	9	3	4	12	88%	
		Day3	78	9	0	3	12	88%	
		Day4	73	12	3	0	9	88%	
		Day5	68	12	2	1	9	88%	
		Day6	68	12	1	3	8	83%	
		Day7	71	10	0	2	12	92%	
	HP2-Winter	Day1	48	10	2	1	11	88%	63%
		Day2	46	10	3	0	11	88%	
		Day3	47	4	6	7	7	46%	
		Day4	46	7	5	6	6	54%	
		Day5	44	6	6	5	7	54%	
		Day6	49	9	2	9	4	54%	
		Day7	52	10	0	10	4	58%	
	HP2-Spring/ Fall	Day1	65	3	5	6	10	54%	53%
		Day2	65	8	7	4	5	54%	
		Day3	69	5	7	4	8	54%	
		Day4	73	9	5	1	9	75%	
		Day5	67	6	2	7	9	63%	
		Day6	65	8	0	13	3	46%	
		Day7	62	5	1	17	1	25%	
	HP4-Summer	Day1	72	13	1	1	9	92%	95%
		Day2	64	13	1	1	8	88%	
		Day3	63	13	0	2	9	92%	
		Day4	67	14	0	1	9	96%	
		Day5	66	14	0	1	9	96%	

Building	Test code	Day	$T_{OD, avg}$ °F	Unoccpancy/ $Unoccpancy_{pred}$	Unoccpancy/ $Occpancy_{pred}$	Occpancy/ $Unoccpancy_{pred}$	Occpancy/ $Occpancy_{pred}$	Accuracy	Weekly Accuracy
		Day6	64	24	0	0	0	100%	
		Day7	71	24	0	0	0	100%	
	HP4-Winter	Day1	48	14	8	2	0	58%	34%
		Day2	46	12	8	4	0	50%	
		Day3	47	8	4	8	4	50%	
		Day4	46	0	3	16	5	21%	
		Day5	44	5	4	11	4	38%	
		Day6	49	5	0	19	0	21%	
		Day7	52	0	0	24	0	0%	
	HP4-Spring/ Fall	Day1	57	13	0	2	9	92%	94%
		Day2	60	13	0	2	9	92%	
		Day3	60	13	0	2	9	92%	
		Day4	64	13	0	2	9	92%	
		Day5	61	13	0	2	9	92%	
		Day6	63	24	0	0	0	100%	
		Day7	64	24	0	0	0	100%	
	HP5-Summer	Day1	88	14	2	0	8	92%	95%
		Day2	88	14	1	0	9	96%	
		Day3	78	15	1	0	8	96%	
		Day4	73	14	0	2	8	92%	
		Day5	68	14	0	0	10	100%	
		Day6	68	21	0	3	0	88%	
		Day7	71	24	0	0	0	100%	
	HP5-Winter	Day1	48	11	0	4	9	83%	85%
		Day2	46	9	1	4	10	79%	
		Day3	47	15	2	2	5	83%	
		Day4	46	9	1	6	8	71%	
		Day5	44	10	2	4	8	75%	
		Day6	49	24	0	0	0	100%	
		Day7	52	24	0	0	0	100%	
	HP5-Spring/ Fall	Day1	65	9	4	6	5	58%	67%
		Day2	65	9	3	7	5	58%	
		Day3	69	7	4	8	5	50%	
		Day4	73	6	1	10	7	54%	
		Day5	67	12	0	6	6	75%	
		Day6	65	21	0	3	0	88%	
		Day7	62	21	0	3	0	88%	
	HP6-Summer	Day1	88	13	4	0	7	83%	91%
		Day2	88	14	3	0	7	88%	
		Day3	78	15	0	0	9	100%	
		Day4	73	12	0	2	10	92%	
		Day5	68	11	0	3	10	88%	
		Day6	68	21	0	3	0	88%	
		Day7	71	24	0	0	0	100%	
	HP6-Winter	Day1	48	8	3	5	8	67%	68%
		Day2	46	7	4	6	7	58%	
		Day3	47	5	4	8	7	50%	
		Day4	46	4	1	10	9	54%	
		Day5	44	6	4	8	6	50%	
		Day6	49	24	0	0	0	100%	
		Day7	52	24	0	0	0	100%	
	HP6-Spring/ Fall	Day1	65	11	0	3	10	88%	88%
		Day2	65	11	0	3	10	88%	
		Day3	69	12	0	2	10	92%	
		Day4	73	14	1	1	8	92%	
		Day5	67	9	0	5	10	79%	
		Day6	65	21	0	3	0	88%	
		Day7	62	22	0	2	0	92%	
	HP7-Summer	Day1	72	12	0	1	11	96%	92%
		Day2	64	12	0	1	11	96%	
		Day3	63	10	0	3	11	88%	
		Day4	67	12	0	2	10	92%	
		Day5	66	12	0	4	10	92%	
		Day6	64	23	0	1	0	96%	

Building	Test code	Day	$T_{OD, avg}$ °F	Unoccpancy/ $Unoccpancy_{pred}$	Unoccpancy/ $Occpancy_{pred}$	Occpancy/ $Unoccpancy_{pred}$	Occpancy/ $Occpancy_{pred}$	Accuracy	Weekly Accuracy
		Day7	65	21	0	3	0	88%	
	HP7-Winter	Day1	48	8	8	0	8	67%	70%
		Day2	46	9	5	3	7	67%	
		Day3	47	8	6	6	4	50%	
		Day4	46	8	4	5	7	63%	
		Day5	44	6	6	7	5	46%	
		Day6	49	24	0	0	0	100%	
		Day7	52	24	0	0	0	100%	
	HP7-Spring/ Fall	Day1	57	8	3	6	7	63%	58%
		Day2	61	8	5	6	5	54%	
		Day3	60	8	4	6	6	58%	
		Day4	64	8	6	6	4	50%	
		Day5	61	9	3	4	8	71%	
		Day6	63	13	0	11	0	54%	
		Day7	64	14	0	10	0	58%	
	HP8-Summer	Day1	88	12	3	0	9	88%	89%
		Day2	88	13	2	1	8	88%	
		Day3	78	15	0	0	9	100%	
		Day4	73	13	0	3	8	88%	
		Day5	68	12	0	4	9	88%	
		Day6	68	21	0	3	0	88%	
		Day7	71	21	0	3	0	88%	
	HP8-Winter	Day1	48	8	8	0	8	67%	57%
		Day2	46	8	4	4	8	67%	
		Day3	47	8	4	6	6	58%	
		Day4	46	8	3	5	8	67%	
		Day5	44	6	5	7	6	50%	
		Day6	49	22	0	2	0	92%	
	HP8-Spring/ Fall	Day1	65	13	0	1	10	96%	94%
		Day2	65	14	0	0	10	100%	
		Day3	69	13	1	0	10	96%	
		Day4	73	14	0	1	9	96%	
		Day5	67	13	0	1	10	96%	
		Day6	65	21	0	3	0	88%	
		Day7	62	21	0	3	0	88%	
Building D	RTU1-summer	Day1	68.2	0	12	0	12	50%	80%
		Day2	71.3	0	8	2	14	58%	
		Day3	75.4	0	0	5	19	80%	
		Day4	75.1	0	0	0	24	100%	
		Day5	75.4	0	0	4	20	83%	
		Day6	73.7	0	1	1	22	92%	
		Day7	68.1	0	0	0	24	100%	
	RTU1-spring/fall	Day1	47.0	0	24	0	0	0%	51%
		Day2	58.4	0	12	0	12	50%	
		Day3	60.5	0	8	0	16	67%	
		Day4	59.2	1	9	0	14	63%	
		Day5	62.0	0	10	1	13	54%	
		Day6	64.7	0	8	0	16	67%	
		Day7	62.0	0	10	0	14	58%	
	RTU2-summer	Day1	73.0	0	0	0	24	100%	93%
		Day2	67.8	0	1	2	21	88%	
		Day3	67.6	0	7	2	15	63%	
		Day4	71.6	0	0	0	24	100%	
		Day5	77.6	0	0	0	24	100%	
		Day6	70.7	0	0	0	24	100%	
		Day7	72.1	0	0	0	24	100%	
	RTU2-spring/fall	Day1	47.7	0	24	0	0	0%	51%
		Day2	58.9	0	12	0	12	50%	
		Day3	61.3	0	8	0	16	67%	
		Day4	59.2	0	10	0	14	58%	
		Day5	62.5	0	10	1	13	54%	
		Day6	65.2	0	8	0	16	67%	
		Day7	62.0	0	9	0	15	63%	
		Day1	71.0	0	4	0	20	83%	74%

Building	Test code	Day	$T_{OD, avg}$ °F	Unoccpancy/ $Unoccpancy_{pred}$	Unoccpancy/ $Occpancy_{pred}$	Occpancy/ $Unoccpancy_{pred}$	Occpancy/ $Occpancy_{pred}$	Accuracy	Weekly Accuracy
Building E	RTU1-summer	Day2	70.8	1	7	2	14	63%	
		Day3	72.9	0	9	2	13	54%	
		Day4	72.9	0	7	2	15	63%	
		Day5	72.3	0	1	1	22	92%	
		Day6	79.2	0	6	1	17	71%	
		Day7	82.3	0	0	1	23	96%	
	RTU1-winter	Day1	44.1	7	2	1	14	88%	88%
		Day2	44.2	7	2	2	13	83%	
		Day3	45.7	7	2	1	14	88%	
		Day4	46.8	7	0	2	15	92%	
		Day5	44.6	7	0	3	14	88%	
		Day6	47.3	9	0	2	13	92%	
		Day7	50.2	8	0	3	13	88%	
	RTU1-spring/fall	Day1	56.2	10	0	11	3	54%	67%
		Day2	56.9	6	5	4	9	63%	
		Day3	61.8	4	4	4	12	67%	
		Day4	65.5	3	3	5	13	67%	
		Day5	60.1	3	3	6	12	63%	
		Day6	61.4	6	0	5	13	80%	
		Day7	61.0	7	0	5	12	80%	
	RTU2-summer	Day1	78.7	0	0	0	24	100%	70%
		Day2	72.3	1	0	4	19	83%	
		Day3	71.7	0	0	0	24	100%	
		Day4	70.8	1	9	1	13	58%	
		Day5	70.7	0	1	0	23	96%	
		Day6	71.4	0	10	2	12	50%	
		Day7	70.5	0	2	3	1	4%	
	RTU2-winter	Day1	51.6	6	7	3	8	58%	70%
		Day2	49.7	6	6	3	9	63%	
		Day3	51.3	6	6	2	10	67%	
		Day4	54.3	5	5	3	11	67%	
		Day5	54.7	6	5	2	11	71%	
		Day6	55.9	7	2	4	11	75%	
		Day7	63.0	7	1	2	14	88%	
	RTU2-spring/fall	Day1	68.9	5	2	1	16	88%	82%
		Day2	68.0	3	2	5	14	71%	
		Day3	62.8	2	1	7	14	67%	
		Day4	56.7	8	3	1	12	83%	
		Day5	56.6	8	2	0	14	92%	
		Day6	55.5	11	3	0	10	88%	
		Day7	55.8	11	3	0	10	88%	
Building F	RTU1-summer	Day1	83.1	10	0	0	14	100%	92%
		Day2	84.1	9	1	2	12	88%	
		Day3	84.4	9	1	2	12	88%	
		Day4	83.0	9	1	1	13	92%	
		Day5	82.5	9	2	1	12	88%	
		Day6	83.0	8	0	2	14	92%	
		Day7	86.3	11	0	1	12	96%	
	RTU1-winter	Day1	64.2	1	0	9	14	63%	85%
		Day2	73.0	7	0	3	14	88%	
		Day3	75.3	8	0	2	14	92%	
		Day4	75.3	8	0	2	14	92%	
		Day5	63.3	7	0	3	14	88%	
		Day6	65.5	7	0	4	13	83%	
		Day7	71.0	9	0	3	12	88%	
	RTU1-spring/fall	Day1	82.7	7	0	3	14	88%	89%
		Day2	80.1	8	0	2	14	92%	
		Day3	83.1	8	0	2	14	92%	
		Day4	82.0	7	0	3	14	88%	
		Day5	81.8	7	0	3	14	88%	
		Day6	82.4	8	0	2	14	92%	
		Day7	82.4	10	0	3	11	88%	
	RTU2-summer	Day1	82.2	8	0	2	14	92%	91%
		Day2	77.0	8	0	3	13	88%	

Building	Test code	Day	$T_{OD, avg}$ °F	Unoccpancy/ $Unoccpancy_{pred}$	Unoccpancy/ $Occpancy_{pred}$	Occpancy/ $Unoccpancy_{pred}$	Occpancy/ $Occpancy_{pred}$	Accuracy	Weekly Accuracy
		Day3	83.1	8	0	2	14	92%	
		Day4	85.0	8	0	2	14	92%	
		Day5	86.0	8	0	2	14	92%	
		Day6	86.9	8	0	2	14	92%	
		Day7	85.2	11	0	2	11	92%	
	RTU2-winter	Day1	64.1	7	3	0	14	88%	91%
		Day2	73.6	9	1	1	13	92%	
		Day3	75.3	9	0	1	14	96%	
		Day4	71.3	9	0	1	14	96%	
		Day5	62.8	9	0	1	14	96%	
		Day6	65.2	8	2	2	12	83%	
		Day7	70.3	10	2	1	11	88%	
	RTU2-spring/fall	Day1	80.4	8	0	2	14	92%	92%
		Day2	77.8	8	0	2	14	92%	
		Day3	80.2	8	0	2	14	92%	
		Day4	80.7	8	0	2	14	92%	
		Day5	77.6	8	0	2	14	92%	
		Day6	79.0	8	0	2	14	92%	
		Day7	78.7	11	0	2	11	92%	

Distribution

No. of Copies

Name
Organization
Address
City, State and ZIP Code

Organization
Address
City, State and ZIP Code
Name
Name
Name
Name
Name (#)

Name
Organization
Address
City, State and ZIP Code

No. of Copies

 # Foreign Distribution

Name
Organization
Address
Address line 2
COUNTRY

Local Distribution
Pacific Northwest National Laboratory

Name	Mailstop
Name	Mailstop
Name	Mailstop
Name	Mailstop
Name	(PDF)

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov