2016

Assisted Point Mapping to Enable Cost-effective Deployment of Intelligent Building Applications

Francesco Leonardi
United Technologies Research Center, United States of America, leonarf@utrc.utc.com

Hayden Reeve
United Technologies Research Center, United States of America, reevehm@utrc.utc.com

Timothy Wagner
United Technologies Research Center, United States of America, wagnertc@utrc.utc.com

Ziyou Xiong
United Technologies Research Center, United States of America, xiongz@utrc.utc.com

June Park
Carnegie Mellon University, United States of America, juneyoup@andrew.cmu.edu

Assisted Point Mapping to Enable Cost-effective Deployment of Intelligent Building Applications

Francesco LEONARDI[1]*, Hayden M. REEVE[2], Timothy C. WAGNER[2], Ziyou XIONG[2], June W. PARK[3]

[1]United Technologies Research Center,
Berkeley, California, USA,
LeonarF@utrc.utc.com

[2]United Technologies Research Center,
East-Hartford, Connecticut, USA,
ReeveHM@utrc.utc.com, WagnerTC@utrc.utc.com

[3]Center for Building Performance and Diagnostics, Carnegie Mellon University,
Pittsburgh, Pennsylvania, USA,
juneyoup@andrew.cmu.edu

* Corresponding Author

ABSTRACT

Over the past years intelligent building applications that promise to dramatically reduce energy consumption, improve occupant comfort and streamline maintenance have been proposed. However their adoption has met a steep barrier in the unexpectedly high cost of mapping data from building automation systems into these applications' data models. In fact the industry does not have a common convention on how to name points. Generally, names are correlated with the semantic of the variables they represent, but typically engineers have the freedom to set up variable names according to their preferences. In the last few years the research community has devoted increasing attention in automating the process of mapping data points from existing BAS. In order to meet market requirements, an "ideal" algorithm must be accurate, able to infer complex relationships between data points, easy to use, and scalable. However, none of the published work meets all the market requirements. Most need an expert user, some are not "easy to use" and none can automatically infer complex relationships.

This paper presents a novel algorithm to tackle the mapping problem. This work contributes to the state-of-the-art by providing an algorithm that can be successfully applied to various datasets with minor or no modification to the algorithm (i.e. no expert in the loop). This is also the first to automatically infer complex relationships with only point names as input (i.e. ease of use).

The algorithm was tested against a large and diversified dataset comprising points from five buildings, two vendors, three distributors and more than 20K points. The algorithm correctly mapped over 90% of the points required by a test application and successfully identified over 90% of VAVs and 80% of AHUs.

1. INTRODUCTION

There is an increasing recognition within both industry and academia (Project Haystack; Bhattacharya, Ploennigs, & Culler, 2015) that semantic metadata associated to points can reduce the maintenance cost of BAS and make software applications more portable across BAS, buildings and vendors. Semantic metadata captures knowledge about points, equipment and buildings. They describe information such as: point types (e.g. is a point a sensor, set-point, control parameter, trend, alarm, etc.), the physical quantity measured/controlled (e.g. temperature, velocity, pressure, etc.), relationship with the physical world (e.g. is the sensor is measuring air, water, etc.), relationship within the system/equipment where the point belongs to (e.g. return air, return water, etc.), relationship with the system/equipment (e.g. to which equipment a point belongs to), relationship between equipment (e.g. hierarchy of an HVAC system, relationship between the physical world and the equipment/system: e.g.: the location of the sensor (which room in a building), list of zones served by an AHU and physical relationship between points (e.g. distance between sensor/actuators).

The process of manually annotating points with semantic metadata is costly and error prone. Typical BASs have thousands of points and would need several times more metadata. A trained field engineer would have to spend many hours to fully annotate a BAS and this is cost prohibitive in many cases. This is the major driving force in the research for techniques and algorithms to assist in the generation of metadata from data points.

Even though most of the knowledge required to assign the correct metadata is not explicitly represented in existing BAS, several algorithms have recently been developed to infer it by analyzing data point values and properties (such as point names).

In order to meet market requirements, an "ideal" algorithm to infer semantic metadata would have five properties: 1) high accuracy and especially low false positive rate (i.e. small number of erroneously mapped points); 2) ability to infer complex relationships between data points (e.g. grouping points by equipment, classifying equipment, establishing relationship between equipment); 3) ease of use: it should leverage only readily available knowledge and data about the system (i.e. not requiring the installation of additional sensors); 4) minimal need of having a domain expert using it; and 5) scalability: the algorithm infers semantic metadata across a variety of BAS, buildings, and vendors with satisfactory results at most requiring minimum human involvement in adjusting settings and parameters. Any approach requiring extensive human effort faces a steep market resistance.

1.1 State of the Art

The last few years witnessed an increasing interest from academia in automatically inferring semantic metadata from points. The algorithm described in (Schumann *et al*., 2014) and in (Schumann *et al*., 2015) learns how to map point names to a prebuilt dictionary from a set of example point names presented by an operator. This algorithm assumes the existence (and the knowledge) of a naming convention of point names. The algorithm attempts to match each point with a definition in the dictionary. The richer the dictionary and its definitions the easier it is to find the right match.

An alternative technique is presented in (Bhattacharya *et al*., 2014). It iteratively learns the naming schema of point names and associates this schema to a common semantic model like Haystack (Project Haystack). Similar to the previous approach the algorithm generates questions (or examples) that a user needs to answer. A characteristic of this solution is that the operator has to provide the "translation" between the naming schema and the semantic model. Therefore this approach is well suited for buildings with limited naming schemas. The problem of "transfer learning" i.e. the ability to learn a model in one or few building and to reuse the same model across several buildings to classify/label points has been addressed only recently. A promising approach is presented in (Hong *et al*., 2015) where the algorithm combines both sensor readings and point names to learn a classifier that can be applied to other buildings. While results are encouraging in labelling point types there is no attempt in inferring relationships between points. Similar conclusions can be drawn from (Balajiy *et al*., 2015) where an active learning algorithm can achieve very high accuracy in establishing point types. In the same paper it was also noted that for many practical uses, identifying point types is not sufficient. Invaluable information such as identifying equipment types and grouping points per equipment can lead to a drastic simplification in the deployment of building applications. The algorithm illustrated in this paper addresses this last problem.

2. Algorithm Description

The primary intent of this inference algorithm is to reduce the effort in deploying intelligent applications on top of existing building automation systems by assisting in the laborious task of mapping embedded controller data to application data.

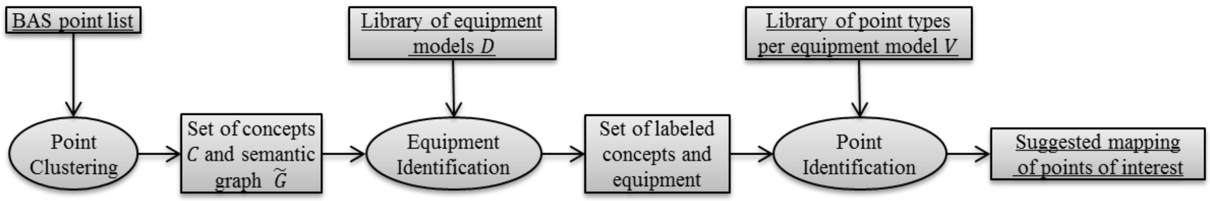

Figure 1: Block diagram of the inference algorithm.

The algorithm digests a list of point names from a building automation system as input and generates suggestions of the likely meaning of points as output. The overall algorithm leverages the fact that point names are usually correlated with the semantic of the variables they represent. The inference algorithm can be logically divided into three steps: Point Clustering, Equipment Identification, and Point Identification (Figure 1). The algorithm takes in a list of point names from a building automation system, a library D of equipment models, and a library V of point types per equipment model. Models are created offline by domain experts or by learning from samples in a training set (as accomplished in this paper). Customized libraries that take into consideration particular characteristics of vendors, distributors and regions can be adopted, trading portability for enhanced accuracy of the algorithm.

2.1 Point Clustering

In this unsupervised step the algorithm first transforms each point name into a list of its constitutive parts (i.e. tokens). Tokens in point names are usually delimited by special characters (e.g.: punctuation, '/', '\', '-') or other tokens. Numbers are also considered as tokens. After this initial step, points are represented in a directed graph $G = \{N, E\}$ where each node $n \in N$ is a token. A directed edge $e_{ij} \in E$ between two nodes (n_i, n_j) indicates that the nodes belong to at least one point and n_i strictly precedes n_j in those point names. The set of edges originating in node n_i is denoted by E_i^- and $\|E_i^-\|$ represents the cardinality of this set. Set $F \subseteq \mathrm{N}$ contains leaf nodes: i.e. nodes representing the last tokens in point names. Each path starting from a source node $n_s \in N$ (i.e. a node without edges pointing to it) and ending on a $n_f \in F$ uniquely identifies a point name.

We introduce the function $T(e)$ that returns the label of the token of the sink node of edge e. We define the equivalence function of two edges as:

$$\epsilon(e_{ij}, e_{nm}) = \begin{cases} 1\ if\ T(e_{ij}) = T(e_{nm}) \\ 0\ otherwise \end{cases} \tag{1}$$

The similarity metric of two nodes is defined as:

$$S(n_i, n_j) = \begin{cases} 1\ if\ \dfrac{0.5}{max(\|E_i^-\|, \|E_j^-\|)} \displaystyle\sum_{e \in E_i^-, \mathrm{f} \in E_j^-} \epsilon(e, f) \geq \vartheta \\ 0\ if\ \|E_i^-\| \leq 2\ or\ \|E_i^-\| \leq 2 \\ 0\ otherwise \end{cases} \tag{2}$$

where $\vartheta \in [0,1]$ is a parameter. When $S(n_i, n_j) = 1$ the two nodes are considered similar and placed in the same cluster. Similar nodes are likely to belong to the same equipment type (e.g. VAVs in a building are likely to have more or less the same point names encoded in similar naming conventions) or other logical groups (e.g. list of equipment, subcomponent of equipment etc.).

The algorithm proceeds by transforming clusters into concepts. A concept is an abstraction of a cluster: it contains the list of tokens that are common across nodes in the cluster. Nodes in a cluster are also instances of the concept of the cluster. Concepts are organized in the semantic graph $\tilde{G}(C, R)$, where C is the set of concepts and R is the set of relationships between concepts. For the purpose of this paper, only the inclusion relationship is relevant. This type of relationship is inferred from C and G. A concept c_i is included by c_j if there is a directed path connecting at least one instance of c_j to one instance of c_i.

2.2 Equipment Identification

The second step of the inference algorithm attempts at automatically labeling (i.e., assigning semantic meaning to the concepts in C. In other words, the algorithm establishes whether a concept represents HVAC equipment and the label of the equipment (e.g. VAV, AHU, etc.). This is accomplished in two separate stages. Stage one considers each individual concept; stage two leverages global knowledge of concepts present in the semantic graph $\tilde{G}$.

2.2.1 Stage one: stage one starts by extracting features from the information present in the concepts (i.e. tokens) and presented in a format that is readily consumed by off-the-shelf machine-learning algorithms. Features are computed by comparing concepts with equipment models D. Each equipment model $M_k \in D$ consists of a set of model-points: i.e. points that are typically present in that particular equipment type. Each model-point $m \in M_k$ is identified by a label expressed in a formal semantic (e.g. Haystack tags, custom tags, etc.) and carries a set of possible alternative token representations that are usually found in the field for that point. To increase classification robustness, likely variations of tokens (e.g., by removing vowels) are also included. The feature for concept j and equipment model i is computed according to:

$$f_{ij} = \left(\beta \sum_{m \in M_i} \max_{p \in c_j}\{G(m,p)\}\right)^2 \tag{3}$$

Where p is a point in concept j, β is a parameter ($\beta = 3$ in this paper) and

$$G(m,p) = \max_{a \in A(m)} \left\{\frac{\sum_l \sum_n d_j(t_l^a, t_n^p)}{\max(\|p\|, \|a\|)}\right\} \tag{4}$$

Where d_j is the Jaro–Winkler distance, a is an alternative token representation for model point m, $A(m)$ returns the alternatives for the m-th model point, t_l^a is the l –th token in the alternative a, and t_n^p is the n-th token of concept point p. $\|p\|$ and $\|a\|$ indicate the number of tokens in concept-point p and alternative a respectively. In general, any off-the-shelf classifier can be used to assign a label to a concept. In particular the results in this paper are obtained with a linear Support Vector Machine (SVM) classifier.

2.2.2 Stage two: the output of stage one can be further refined by leveraging global knowledge of the relationships between concepts present in the semantic graph $\tilde{G}$. In particular new types of concepts (e.g. collections) can be only labeled at this stage. For example a concept including only concepts of one type can now be labeled as a collection. Similarly some classes of erroneous classifications generated at stage one can now be corrected. Two examples: a VAV is unlikely to include an AHU and a concept including both types of equipment is likely to be neither of them.

2.3 Point Identification

The final step in the algorithm consists in identifying points in labeled concepts. In order to achieve this, the algorithm finds for each point in a labeled concept the best matching point in the corresponding equipment-model in V. Library V is derived from D and it can also include additional model-points that, while less common, might still be useful for some intelligent applications. The label of each point p in a labeled concept c_i is determined by:

$$L\left(M_l^v, \underset{m \in M_l^p}{\operatorname{argmax}}\{G(m,p)\}\right) \tag{5}$$

Where M_l^v represents the model in V of c_i and $L(M_l^v, n)$ returns the label of the m-th model-point in M_l^v .

3. Results

This section demonstrates the performance of the algorithm on a large dataset comprising data points from 9 buildings with BASs from multiple vendors and dealers.

Table 1: Dataset composition and cluster results.

Building		#BAS points	#C	#Points in $\tilde{G}$	#Un-clustered points
ID	Vendor				
1	JCI	1160	26	126 (10%)	0
2	Siemens	4308	36	145 (3.3%)	121 (2.8%)
3	JCI	3545	50	329 (9.3%)	1
4	ALC	5015	93	432 (8.6%)	200 (4.0)
5	Siemens	1808	29	101 (5.6%)	82 (4.5%)
6	Siemens	5981	49	213(3.6%)	119(2.0%)
7	Siemens	8247	38	205 (2.5%)	321 (3.9%)
8	ALC	3448	44	211 (6.1%)	53 (1.5%)
9	ALC	3428	33	143 (4.2%)	85 (2.5%)

Table 2: Equipment identification results.

ID	#VAV			#AHU		
	GT	FP	FN	GT	FP	FN
1	107	0	0	8	0	1
2	51	1	1	-	-	-
3	148	0	1	6	0	0
4	101	0	0	7	2	0
5	181	0	16	6	0	0
6	82	0	5	-	-	-
7	421	3	0	15	3	0
8	52	3	0	2	0	0
9	100	0	0	1	0	1

3.1 Point Clustering

Table 1 summarizes the point clustering results. The first three columns report the building ID, vendor name, and the number of points in the building. Column 4 lists the number of concepts obtained by the algorithm. Column five represents the number of points in the semantic graph $\tilde{G}$. The last column summarizes the number of points that the algorithm is not able to assign to any cluster. The number of points per building ranges from about 1000 to 8000. The dataset is divided in a training set (building 1 to 4) and a test set (building 5 to 9). The first set was used to create library models and tune parameters. The second set was used to assess the algorithm performance. The test set consists of data points collected from 5 buildings with 2 BMS vendors and 3 dealers.

Depending on the considered building, the number of clusters identified by the algorithm ranges from 26 to 93. More importantly the number of points in the semantic graph $\tilde{G}$ is on the order of 3 to 7% of the total number of points and the number of un-clustered points is less than 5% of the total number of points. This means that by assigning a semantic definition to each concept and to each concept-point it is possible to annotate most of the dataset. In other words clustering by itself can drastically reduce the effort required to label a dataset.

3.2 Equipment Identification

Equipment identification results are shown in Table 2. The first columns represents the building ID, columns 2 and 5 report the ground truth (GT) of the number of VAV and AHU per building. Columns 3 and 4 summarize the number of false positives (FP) and false negatives (FN) of VAV classified by the algorithm. Columns 6 and 7 list similar results for the AHUs. The algorithm accurately recognizes more than 90% of the VAVs and more than 80% of the AHUs present in a building. It is worth mentioning that the classifier was unable to detect the only AHU of building 9. This happened because of a current limitation of the clustering algorithm that requires the presence of at least two similar groups of points/equipment in order to create a cluster. In this case, having only one AHU, the clustering algorithm does not create a cluster for the AHU points and therefore the classification algorithm is unaware of it. In general it appears that correctly classifying AHUs is more difficult than for VAVs. A possible explanation is that the training set has fewer examples of AHUs and this makes it more challenging to generalize a robust classifier.

Table 3: Point Classification VAV

ID	Sns Dmpr Position			Sns Flow Supply Air			Air Flow Supply SP			Sns Zone Temp		
	GT	FP	FN	GT	FP	FN	GT	FP	FN	GT	FP	FN
1	107	0	0	107	0	0	-	-	-	107	0	0
2	51	0	1	51	0	1	51	0	1	51	1	1
3	148	0	1	148	0	1	148	0	1	144	-	5
4	101	0	0	101	0	0	101	0	0	4	0	0
5	-	-	-	181	0	16	-	-	-	181	0	16
6	82	0	5	82	0	5	82	0	5	82	0	5
7	421	3	0	421	3	0	421	3	0	421	3	0
8	52	3	0	52	3	0	52	3	0	21	0	0
9	-	-	-	100	0	0	100	0	0	98	0	0

Table 4: Point Classification AHU

ID	Sns Rtrn Air Temp			Sns Mixed Air Temp			Outside Air Dmpr			Sns Rtrn Air Hum			Sns Outside Air Temp		
	GT	FP	FN	GT	FP	FN	GT	FP	FN	GT	FP	FN	GT	FP	FN
1	8	0	0	8	0	0	8	0	0	8	0	0	8	0	0
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	2	-	-	6	0	0	6	0	0	2	4	0	-	-	-
4	7	2	0	7	2	0	7	0	0	7	-	-	-	-	-
5	6	0	0	-	-	-	-	-	-	-	-	-	6	0	0
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	15	0	0	15	0	0	-	-	-	15	0	0	-	-	-
8	2	0	0	2	0	0	-	-	-	1	0	1	-	1	-
9	1	0	1	1	0	1	1	0	1	1	0	1	-	-	-

3.3 Point Identification

Point identification results for VAVs and AHUs are reported in Table 3 and 4 respectively. The first row of both tables indicates some of the supported point types expressed with custom semantic tags. For each point type three columns summarize the number of occurrences of the point (GT), the number of false positives (FP) and false negatives (FN) identified by the algorithm. In most cases, the algorithm classifies points with more than 90% accuracy. Similar to the equipment identification, building 9 represents an exception. Since a cluster was not created, it is not possible to detect any of the AHU points for building 9. Additional work is required to enable the algorithm to properly identify clusters and points for buildings with single pieces of equipment.

6. PRELIMINARY COST MODEL

In order to truly understand the impact of any tool for semantic inference of BAS points, a cost model for mapping data points from a BAS into an application must be introduced. Assuming that an operator requires a constant amount of time to assign each semantic tag, a simplified cost model can be written as a function of the number of tags (τ) that need to be assigned:

$$\varphi(\tau) = \|B\|\tau_B + \sum_{b\in B}\sum_{e\in E} I(e,b)\left(\tau_e + \mathrm{P(e)}\tau_{\mathrm{p}}\right) \tag{6}$$

Where B represents a set of buildings, τ_B is the number of tags that are specific to a building (examples of building tags are name, location, area, type, climate zone, etc.), E is the set of equipment types that are of interest of the application, function $I(e,b)$ returns the number of instances of an equipment type e in a building b, τ_e is the number of tag per equipment type (examples of equipment tags are type, location, relationship with zones and other equipment, etc.), function P(e) returns the number of points in equipment type e, and τ_p is the number of tag per point (examples of point tags are: point definition and reference to containing equipment).

The first term of the cost model ($\|B\|\tau_B$) is usually small compared to the second one. In fact in a typical campus there are many more pieces of equipment and data points than buildings and $\tau_B \approx \tau_e$. It is understandable that the research presented so far in the open literature attempts to minimize the number of point tags that need to be manually assigned. In particular most activities are directed towards automatically inferring point definition tags. A few papers present results on establishing the location of sensors and relationships of equipment. The algorithm described in these pages is the first introduction of a viable mechanism to automatically infer both point definition tags, references to containing equipment and equipment tags.

7. CONCLUSIONS

This paper presented a novel algorithm for inferring semantics of BAS data points. This algorithm is the first to automatically infer complex relationships with only point names as input. It can also successfully operate on data point from various buildings with minor or no modification to the algorithm. Results obtained from a large and diversified dataset comprising points from five buildings, two vendors, three distributors and more than 20K points indicate that the algorithm can correctly map over 90% of the points required by a test application and successfully identified over 90% of VAVs and 80% of AHUs.

While knowing the equipment type and the associated points is important in many applications, there are cases where other relationships are relevant too. In fact, there have been attempts to infer the hierarchy of VAVs and AHUs (Pritoni *et al*, 2015) and spatial location of sensors in (Ortiz *et al*., 2013) and in (Koc *et al*., 2014). Future work should definitely revisit and incorporate these approaches. Future studies might also use additional knowledge about points (e.g. values, measurement units, etc.) to increase the robustness and the accuracy of the output as suggested by (Balajiy, 2015) and (Hong, 2015).

NOMENCLATURE

BMS	Building Management System
BAS	Building Automation System
VAV	Variable Air Valve
AHU	Air Handling Unit
HVAC	Heating Ventilation Air Conditioning
GT	Ground Truth
FP	False Positive
FN	False Negative
Sns Rtrn Air Temp	Data point reporting the return air temperature
Sns Mixed Air Temp	Data point reporting the mixed air temperature
Outside Air Dmpr	Data point reporting the position of a damper for the outside air
Sns Rtrn Air Hum	Data point reporting the return air humidity
Sns Outside Air Temp	Data point reporting the outside air temperature
Sns Dmpr Position	Data point reporting the position of a damper
Sns Flow Supply Air	Data point reporting the air supply flow
Air Flow Supply SP	Data point reporting the value of the air supply flow set-point
Sns Zone Temp	Data point reporting the temperature of a zone

REFERENCES

Balajiy, B., Vermay, C., Narayanaswamyy, B., & Agarwal, Y. (2015). Zodiac: Organizing Large Deployment of Sensors to Create Reusable Applications for Buildings. *Proceedings of the 2nd ACM Conference on Embedded Systems for Energy-Efficient Buildings.*

Bhattacharya, A., Culler, D. E., Ortiz, J., Hong, D., & Whitehouse, K. (2014). *Enabling portable building applications through automated metadata transformation.* University of California: University of California.

Bhattacharya, A., Ploennigs, J., & Culler, D. (2015). Short Paper: Analyzing Metadata Schemas for Buildings: The Good, the Bad, and the Ugly. *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments* (pp. 33-34). ACM.

Hong, D., Wang, H., Ortiz, J., & Whitehouse, K. (2015). The Building Adapter: Towards Quickly Applying Building Analytics at Scale. *Proceedings of the 2nd ACM Conference on Embedded Systems for Energy-Efficient Buildings.*

Koc, M., Akinci, B., & Berges, M. (2014). Comparison of linear correlation and a statistical dependency measure for inferring spatial relation of temperature sensors in buildings. *Proceeding BuildSys '14 Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings.*

Ortiz, J., Dezhi, H., Whitehouse, K., & Culler, D. (2013). Towards automatic spatial verification of sensor placement in buildings. *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings.*

Pritoni, M., Bhattacharya, A. A., Culler, D., & Modera, M. (2015). Short Paper: A Method for Discovering Functional Relationships Between Air Handling Units and Variable-Air-Volume Boxes From Sensor Data. *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments* (pp. 133-136). ACM.

Project Haystack. (n.d.). Retrieved 04 26, 2016, from Project Haystack: http://project-haystack.org/

Schumann, A., & Lécué, F. (2015). Minimizing User Involvement for Accurate Ontology Matching Problems. *Twenty-Ninth AAAI Conference on Artificial Intelligence.*

Schumann, A., Ploennigs, J., & Gorman, B. (2014). Towards automating the deployment of energy saving approaches in buildings. *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings.* ACM.

ACKNOWLEDGEMENT

This material is based upon work supported by the Consortium for Building Energy Innovation, sponsored by the U.S. Department of Energy under Award Number DE-EE0004261. The authors would like also to acknowledge the generous contribution of Intellimation (http://www.intellimation.net/) for providing some of the datasets used in this study.

DISCLAIMER