A Metadata Inference Framework to Provide Operational Information Support for Fault Detection and Diagnosis Applications in Secondary HVAC Systems

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Civil and Environmental Engineering

Jingkun Gao
B.S., Material Chemistry
M.S., Civil Environmental Engineering

Carnegie Mellon University
Pittsburgh, PA
December, 2017

Keywords: FDD, HVAC, BAS, metadata inference.

Abstract

As the cost of hardware decreases and software technology advances, building automation systems (BAS) have been widely deployed to new buildings or as part of the retrofit to replace the old control systems. Though they are becoming more prevalent and promise important benefits to the society, such as improved energy-efficiency and occupants' comfort, many of their benefits remain unreachable. Research suggests that this is because of the heterogeneous, fragmented and non-standardized nature of existing BASs. One of the purported benefits of these systems is the ability to reduce energy consumption through the application of automated approaches such as fault detection and diagnosis (FDD) algorithms. Savings of up to 0.16 quadrillion BTUs per year could be obtained in the US alone through the use of these approaches, which are just software applications running on BAS hardware. However, deployment of these applications for buildings remains a challenge due to the non-trivial efforts of organizing, managing and extracting metadata associated with sensors (e.g., information about their type, function, etc.), which is required by them. One of the reasons leading to the problem is that varying conventions, acronyms, and standards are used to define this metadata. Though standards and government-mandated policies may lift these obstacles and enable these software-based improvements to our building stock, this effort could take years to come to fruition and there are alternative technical solutions, such as automated metadata inference techniques, that could help reign in on the non-standardized nature of today's BASs.

This thesis sheds light on the visibility of this alternative approach by answering three key questions, which are then validated using data from more than 400 buildings in the US: (a) What is the specific operational information required by FDD approaches for secondary heating, ventilation, and air conditioning (HVAC) systems found in existing literature? (b) How is the performance of existing metadata inference approaches affected by changes in building characteristics, weather conditions, building usage patterns, and geographical locations? (c) What is an approach that can provide physical interpretations in the case of incorrect metadata being inferred? We find that: (a) The BAS points required by more than 30% of FDD approaches include six sensors in AHUs monitoring supply air temperature, outside air temperature, chilled water valve position, return air temperature, supply air flow rate, and mixed air temperature; (b) The average performance of existing inference approaches in terms of accuracy is similar across building sites, though there is significant variance, and the expected accuracy of classifying the type of points required by a particular FDD application for a new unseen building is, on average, 75%; (c) A new approach based on physical models is developed and validated on both the simulation data and the real-world data to infer the point types confused by data-driven models with an accuracy ranging from 73% to 100%, and this approach can provide physical interpretations in the case of incorrect inference. Our results provide a foundation and starting point to infer the metadata required by FDD approaches and minimize the implementation cost of deploying FDD applications on multiple buildings.

To my family.

Acknowledgments

First and foremost, I am sincerely grateful for the opportunity to work with Professor Mario Bergés, who has served as the chair of the committee, my advisor, mentor, and friend at Carnegie Mellon University and has been the biggest help during my whole Ph.D. life. I am indebted to him for his great support for all aspects of my professional and personal development. I would also like to thank Professors Burcu Akinci, Barnabás Póczos, Xuesong Liu for valuable suggestions and feedbacks during my thesis proposal and serving as my committee members. This thesis would not have been possible without the support from them. I also want to thank Dr. Youngchong Park, Erik Paulson, and Andrew Boettcher from Johnson Controls for providing the data used in the research. I am thankful to Dr. Michael Brambley, Dr. Sen Huang, Stevens Andrew from Pacific Northwest National Lab for insightful discussions along with the simulation data provided in this work. I also thank my colleagues and friends in the Intelligent Infrastructure Research Lab (INFER Lab) at Carnegie Mellon. You have each helped me in so many ways, discussing research questions, providing me feedbacks, giving me support and encouragement. It has been a true pleasure to work with each of you. I would like to acknowledge Scholarship Council, Siebel Foundation, Advanced Research Projects Agency-Energy (Award Number DE-AR0000705) for the funding that supported my research thesis. Finally, I would like to thank my parents and my wife, for their unconditional love and endless support.

Contents

List of Tables

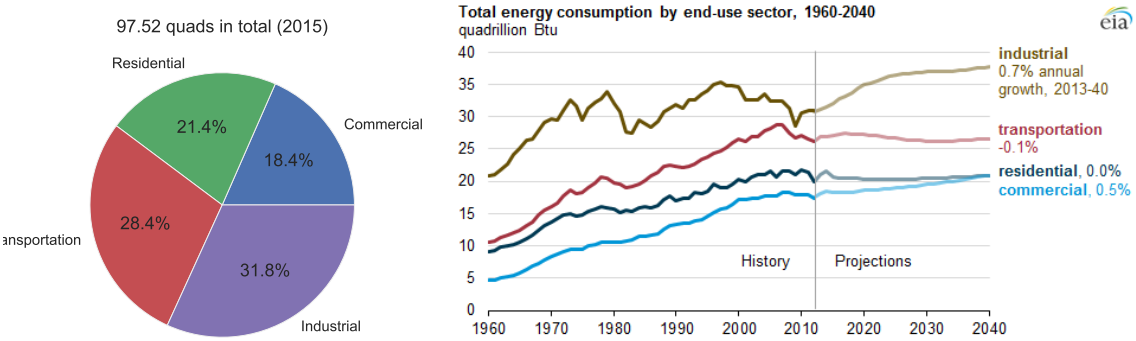
List of Figures

Chapter 1

Introduction

Commercial buildings in the United States (US) consumed 17.97 quadrillions British Thermal Units (quads) in the year 2015, which is equal to 18.4% of the total energy consumed in the US in that year, as seen in Figure 1.1a. This consumption is projected to further increase at an annual rate of 0.5% according to US Energy Information Administration (EIA) [1]. Figure 1.1b shows that the commercial sector will consume as much energy as the residential sector by 2040. The potential savings by improving the energy use in commercial buildings have proven to be sizable according to recent research [2, 3, 4, 5]. For example, in [2] researchers demonstrate 20-30% energy savings could be attributed to re-commissioning of the heating, ventilation, and air conditioning (HVAC) systems to rectify faulty operations, based on a study covering a modest number of commercial office buildings. In [6] and [3] it has been shown that various faults cause one quad of energy waste in commercial buildings, which equals to about 11% of the energy consumed in 2005. Faults in HVAC systems, such as duct leakage, condenser fouling, airflow not balanced, and others, account for more than 80% of this wasted energy. To put these numbers in perspective, one quad is approximately equal to the annual electricity consumption

of Italy [7]. As a result, many fault detection and diagnosis (FDD) approaches have been developed for building HVAC systems to reduce the energy usage by improving the operational performance of commercial buildings [8, 2, 9, 10, 11]. Applications developed based on these approaches do not only reduce energy waste, but also save time and money for building operators to troubleshoot, improve the occupants' comfort (given that people on average spend 87% of their time indoors [12]), lower environment impact, decrease cost for equipment repair and replacement, and many others [2].



(a) Energy consumption by sector (b) Energy projections by sector up to 2040

Figure 1.1: Energy consumption patterns in the United States (Source: US Energy Information Administration)

The function of FDD approaches is to detect any system malfunctions and diagnose the primary causes. These results are then used by building operators to rectify the faults. The detection usually involves comparing the actual performance to a reference [13]. The actual performance is measured based on different sensing points in HVAC systems and the reference could either be calculated based on a physical model using building design information or derived from historical sensing data of normal status. The diagnosis focuses on identifying the magnitude and causes of the faults, as well as isolating their type and location. Facility managers and building operators, who are in charge of maintaining buildings, can benefit from these FDD

tools.

To evaluate the feasibility of implementing FDD approaches in real buildings, EIA enlisted the help of International Energy Agency (IEA) Annex 34 ([2, 5]) to conduct thirty case studies involving twenty-six FDD tools on twenty buildings across twelve countries. One of the conclusions from the study is that the amount of information required by FDD tools and the effort required to extract it should not be overlooked and underestimated. This required information can be classified into main categories (see e.g. [2], [14], and [15]):

- Design information: This category includes the information generated, or gathered during the design phase of the building before it is in operation. Some examples include geometric data like component locations, floor maps structure drawing, HVAC system design data like the manufacturing data for each HVAC equipment (e.g., duct, air handler unit, terminal box), etc.

- Operational information: This category includes the operational phase information when the building starts to function and be occupied. Information associated with sensing, actuating devices, as well as set points monitoring and controlling running status of HVAC systems, falls into this category.

- FDD parameters: This class represents a set consisting of tuning parameters of FDD algorithms, including rule thresholds, model parameters, the window length of the signal, etc.

Design information can often be retrieved from design drawings, spreadsheets, floor maps, diagrams and equipment manuals [16]. FDD parameters are typically chosen either heuristically or based on training data [2]. Operational information is

accessible through a centralized building automation system (BAS)[1] [2, 15], which controls the HVAC system of a building to ensure operational performance and occupants' comfort utilizing various sensing and actuating points. These points inside a BAS are also called **BAS points**, and they represent the sensors and actuators that are distributed throughout the system to monitor and change its state. Typical examples include sensors used to monitor airflow, CO_2, humidity, temperature, and occupancy, as well as actuators associated with valves, dampers, and fans to control the flow of fluid and air.

However, extracting needed information to make it ready to use for FDD tools is not without challenges. Researchers in [17] reported a practical experience of implementing the expert-based FDD systems on three different sites. It was found that the cost of gathering the required information by FDD tools is rather high as the operational information available in a BAS cannot be retrieved automatically. For each test of installing the FDD tool, more than one day of manual effort was spent to extract and standardize the operational information from a BAS. Here the standardization refers to the process of converting operational information extracted from different buildings into a consistent and understandable namespace. In [5] it is also claimed that the cost of running FDD tools is too high due to the difficulty of interfacing with data. Researcher in [18] show that the lack of standardized and structured descriptive data prevents automated FDD algorithms from properly connecting to data without building-specific customization. Additionally,

[1]The centralized system for buildings has many names, including building control system (BCS), building management system (BMS), building energy management system (BEMS), energy management and control system (EMCS), building direct digital control system (DDC), etc. They are same in the sense that they manage and control buildings using sensing and actuating devices, with the minor difference being their perspectives. For example, BEMS focuses more on energy consumption side while BCS focuses more on the control side for HVAC, lighting, and power system.

in [19] it is mentioned that one BAS point will cost approximately one minute to be understood and interpreted, which accumulates to 83 hours of manual investigation for 5,000 points (a typical number of points for a modern medium-sized commercial building). The Department of Energy [20] also concluded that the lack of standard data formats, terms and definitions are significant barriers preventing improvements to building performance. Because various equipment vendors, sensor manufacturers, and controlling components are involved in the setup process of the automation systems, distinct formats, naming conventions and syntaxes are being used to describe data across buildings. Such inconsistency makes it complex to retrieve, manage, understand, integrate and make use of the metadata that describes BAS points [21, 5, 13, 22, 18, 23].

The lack of readily accessible operational information for FDD approaches originates from the design and installation phase when instrumenting buildings, as there is no common standard to follow in terms of defining metadata of BAS points. **Metadata** here refers to the information that helps to identify and contextualize the BAS point, such as the physical quantity it is measuring or changing (e.g., temperature, humidity, flow, pressure), the medium it is interacting (e.g., water, air), the hierarchal physical location (site, building, floor, room, zone, etc.), the unit and range of the measurements, and many others. This metadata is essential to enable effective use of FDD approaches, yet in most situations, it is either unreliable, uninterpretable or outdated [22, 24].

The goal of this research is to develop a metadata inference framework, which provides operational information support to facilitate implementing FDD applications on multiple buildings. One of the main focuses is to tackle the problem of inconsistent metadata associated with BAS points across buildings, which, once

solved, will enable deployment of portable FDD applications [2].

The issue of inconsistent metadata in buildings was encountered as early as 2001 when researchers in [2] were performing case studies implementing FDD application in real buildings, and they concluded that a good point naming convention could lower the cost of implementing FDD tools. As part of the efforts to facilitate exchanging information between participants to test and validate FDD tools on different buildings, a triplet-based standard point-naming scheme was then proposed. Additionally, [22, 18, 25] and [26] also proposed different standards for point names. However, these solutions have not been widely adopted in the existing building stock, and one of the reasons may be that none of these naming schemes cover the wide variety of assets and information dimensions of metadata as is described in [27]. Besides, it is expensive in terms of manual efforts (e.g., one day for one test site) to convert inconsistent names from another building system to a new proposed standard naming space manually [17]. Thus, recent research has focused on semi-automating the process of acquiring and standardizing metadata directly from the information available in BAS using computerized algorithms.

The following section first presents a motivating case study, which explores the difficulties of implementing a rule-based FDD approach from [15] on five different buildings. The industry challenges of large-scale implementations are identified followed by the problem statement. Then a literature review is conducted to identify research gaps, which drive the research scope and research questions that follow. The last section outlines the thesis document organization.

[2]Here portable FDD applications refer to those that are scalable to be deployed on multiple buildings with minimal customized configurations.

1.1 Motivating Case Study

As an initial attempt to understand how to implement FDD approaches in real-world facilities, we conduct a case study to implement a rule-based FDD approach named air handling unit (AHU) performance assessment rules (APAR) [15], which is one of the most highly cited papers discussing FDD in HVAC systems. The rules are applied on AHUs located in five buildings on the campus of Carnegie Mellon University (CMU). To illustrate how the rules are applied to AHUs, the structure of a typical AHU is shown in Figure 1.2. The outside air (OA, lower left) is drawn into the duct due to the pressure difference and then mixed with return air (RA, upper right) through a heat wheel drive to adjust humidity and make use of the free heat in the return air. At the lower left, the damper is used to control the amount of fresh air drawn into the system, and the yellow component serves as an air filter. The mixed air is then conditioned either through the cooling coil or the heating coil to reach the temperature set point for the supply air (SA, lower right). The cooling or heating amount is adjusted by the valve to control how much chilled or hot water is supplied to cycle in the coil. The conditioned air is eventually blown into variable air volume (VAV) boxes above the ceiling of each zone. To balance the air pressure, another fan will blow the exhausted air (EA, upper left) out of the AHU.

To assess the performance of AHUs to be faulty or normal, APAR classifies modes of AHU operation in 1) heating; 2) cooling without mechanical cooling; 3) mechanical cooling with 100% outdoor air; 4) mechanical cooling with minimum outdoor air; and 5) abnormal states which do not fall into any of above four categories.

In mode 1, the outside air damper is positioned to allow the minimum outdoor air fraction necessary to satisfy ventilation during heating; in mode 2, cooling is maintained by adjusting the outside air damper without mechanical cooling, mean-

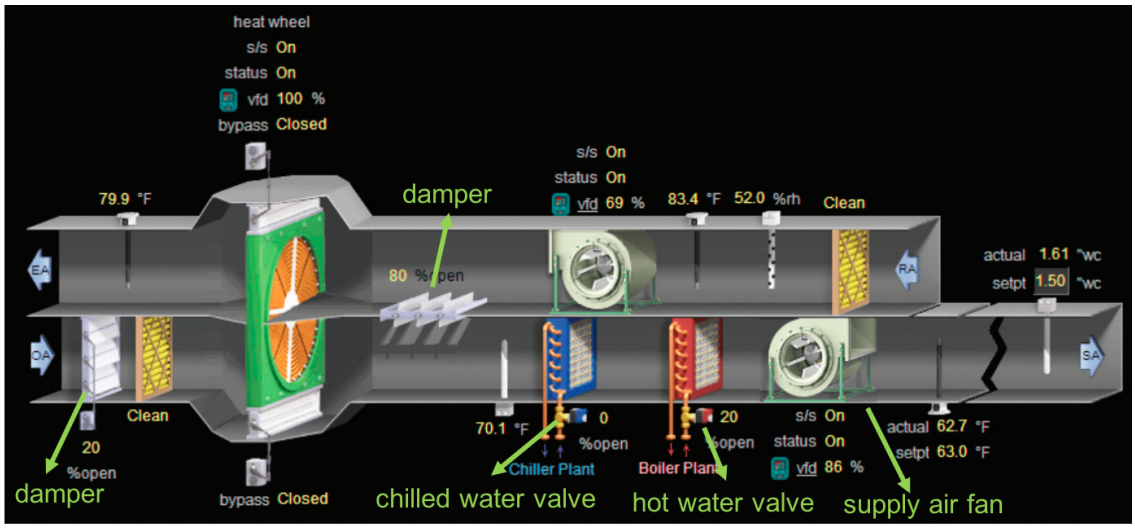


Figure 1.2: Screenshot of one AHU from EIKON-LogicBuilder. It draws outside air (OA) and mixes with return air (RA). Mixed air is conditioned and turned into supply air (SA). Exhausted air (EA) is blown out to balance air pressure

ing the chilled water valve is fully closed; in mode 3 and 4, a comparison between outdoor and return enthalpies is used to make decision either taking 100% outdoor air or minimal outdoor air for mechanical cooling. Other unknown states fall into mode 5.

Under different operation modes, 28 rules are generated to detect and diagnose faults. The faults are categorized into five types:

1. Stuck or leaking mixing box dampers, heating coil valves, and cooling coil valves;

2. Temperature sensor faults;

3. Design faults such as undersized coils;

4. Controller programming errors related to tuning, set points, and sequencing logic;

5. Inappropriate operator intervention.

Table 1.1 lists the required BAS points for deciding modes of operations and evaluating the 28 rules. The last column shows the acronyms for the points being used for APAR. In addition to the points required, some other parameters like thresholds as well as the design phase information (e.g., the minimum outside air damper percentage) are also needed. Among these 13 types of different points, the last two points about the temperature rise across fans are typically chosen heuristically to be 2 F° [15, 14]. As a result, only the remaining 11 types of BAS points need to be extracted from each AHU to detect faults using APAR.

	Description of points	Acronym
Valve	hot water valve	HW VLV
	chilled water valve	CHW VLV
Temperature	mixed air temperature	MAT
	outside air temperature	OAT
	return air temperature	RAT
	supply air temperature	SAT
Set Point	supply air temperature set point	SAT SPT
Humidity	outside air relative humidity	OA RH
	return air relative humidity	RA RH
Damper	outside air damper	OAD
	mixed air damper	MAD
Fan	supply fan temperature rise	ΔSFT
	return fan temperature rise	ΔRFT

Table 1.1: Required points for APAR

A detailed summary of all 28 rules and needed BAS points is shown in Table 1.2. By checking whether the rule is satisfied using historical time series data from needed points, a fault can be detected. Rules 25 to 28 are applied to all modes of operation, and the remaining rules are only applied to the specific mode shown in the first column of the table. In the column of rules, the variables which have either `THRS`, `min`, `max`, Δ in their names are all FDD parameters for the rules, which can be

determined heuristically or tuned based on the operations of actual systems. "COT" represents the change-over outside air temperature when switching between mode 3 and 4. In addition to these variables, the rest are 11 different types of points from a BAS. A check mark represents whether the specific point of that column is needed by the rule in that row.

To apply the same set of 28 rules to AHUs in multiple buildings, we need to map the BAS points from different units to a common standardized namespace, such as acronyms following the same naming convention. Table 1.1 shows one such naming convention. The mapping will simplify the process to apply rules. In other words, once rules are generated based on a particular naming convention, they can be ported directly to any buildings with BAS points following that convention without extra efforts.

We show the details of mapping points for different AHUs in Table 1.3. For each single BAS point in an AHU, we extract the prefix (long string shown in the first row), the point suffix (e.g., "MAT"), point description (additional information to annotate the point, which is normally unavailable) and corresponding time series samples. Notice the prefix and suffix together make up the whole **BAS tag** for this point. By reviewing the combination of all these information and making use of domain knowledge, we identify 11 required points (listed in the left column) for each unit. The one with `NA` represents this required point is not available in this unit and the one with question mark represents we are uncertain about the meaning of this specific point. The `Total` row means the number of total points for this unit in a BAS. The rows with name `NA` and `?` count the number of unavailable and uncertain points. Uncertain points refer to those having obscure descriptive naming tags, for example, the unit in the last column has points named "VRT" and "VRH"

mode	rules	HW VLV	CHW VLV	MAT	OAT	RAT	SAT	SAT SPT	OA RH	RA RH	OAD	MAD
mode 1	1. SAT < MAT + ΔSFT - THRS_t	✓	✓	✓			✓				✓	
	2. if |RAT - OAT| > ΔT_min, |OAF - OAF_min| > THRS_f	✓	✓		✓	✓					✓	
	3. |HW VLV - 1| < THRS_hc & SAT SPT - SAT > THRS_t	✓	✓				✓	✓			✓	
	4. |HW VLV - 1| < THRS_hc	✓	✓								✓	
mode 2	5. OAT > SAT SPT - ΔSFT + THRS_t	✓	✓		✓		✓	✓				
	6. SAT > RAT - ΔRFT + THRS_t	✓	✓			✓	✓					
	7. |SAT - ΔSFT - MAT| > THRS_t	✓	✓	✓			✓					
mode 3	8. OAT < SAT SPT - ΔSFT - THRS_t	✓	✓		✓	✓	✓	✓	✓	✓		
	9. OAT > COT + THRS_t	✓	✓		✓	✓			✓	✓		
	10. |OAT - MAT| > THRS_t	✓	✓	✓	✓	✓			✓	✓		
	11. SAT > MAT + ΔSFT + THRS_t	✓	✓	✓	✓	✓	✓		✓	✓		
	12. SAT > RAT - ΔRFT + THRS_t	✓	✓		✓	✓	✓		✓	✓		
	13. |CHW VLV -1| < THRS_cc & SAT - SAT SPT > THRS_t	✓	✓		✓	✓	✓	✓	✓	✓		
	14. |CHW VLV -1| < THRS_cc	✓	✓		✓	✓			✓	✓		
mode 4	15. OAT < COT - THRS_t	✓	✓		✓	✓			✓	✓		
	16. SAT > MAT + ΔSFT + THRS_t	✓	✓	✓	✓	✓	✓		✓	✓		
	17. SAT > RAT - ΔRFT + THRS_t	✓	✓		✓	✓	✓		✓	✓		
	18. if |RAT - OAT| > ΔT_min, |OAF - OAF_min| > THRS_f	✓	✓		✓	✓			✓	✓		
	19. |CHW VLV -1| < THRS_cc & SAT - SAT SPT > THRS_t	✓	✓		✓	✓	✓	✓	✓	✓		
	20. |CHW VLV -1| < THRS_cc	✓	✓		✓	✓			✓	✓		
mode 5	21. CHW VLV > THRS_cc & HW VLV > THRS_hc & THRS_md < MAD < 1 - THRS_md	✓	✓									✓
	22. HW VLV > THRS_hc & CHW VLV > THRS_cc	✓	✓									
	23. HW VLV > THRS_hc & MAD > THRS_cc	✓										✓
	24. THRS_md < MAD < 1 - THRS_md & CHW VLV > THRS_cc	✓	✓									✓
all	25. |SAT - SAT SPT| > THRS_t						✓	✓				
	26. MAT < min(RAT, OAT) - THRS_t			✓	✓	✓						
	27. MAT > max(RAT, OAT) + THRS_t			✓	✓	✓						
	28. # mode transitions/hour > maximum # mode transitions/hour	✓	✓		✓	✓			✓	✓	✓	

Table 1.2: A summary of 28 rules and points needed, mode of operation depends on the points including HW VLV, CHW VLV, OAT, RAT, OA RH, RA RH, and OAD

which we suspect might be the return air temperature and the return air humidity; however, we do not have enough evidence to support this claim just from tag names and time series data, and we put a question mark before these uncertain points. The symbol ✓ represents the number of usable points for APAR. For each count,

we also calculate the percentage of the number of total points needed (11) inside the parentheses. The rough estimation of time cost indicates that we need at least one minute for one BAS point, which is consistent with what is claimed in [19].

This case study allowed us to understand several difficulties of extracting needed operational information to implement FDD tools in real-world buildings. We summarize them as follows:

1. **Incomplete information**: tags and descriptions do not include complete information to yield meaningful interpretations of type and functionality of BAS points. For example, it is difficult to decipher whether "HCO" or "HCV" should be used as the hot water valve (HW VLV) in the first AHU from Table 1.3. In cases like this, the graphical interface of a BAS or the control logic specifying operation sequence could help to clarify the mappings. However, this constitutes additional manual effort, and the information is not always available.

   The incomplete information will impact the deployment of FDD applications. As is seen in Table 1.4, 25 out of 28 rules are not usable due to two such points. Despite the fact that the number of points with incomplete information is small, the percentage of unusable rules could be very high.

2. **Inconsistent naming conventions**: different tag names are sometimes used for the same point across different buildings, as is seen in Table 1.3. One example is "SAT SPT", which is named differently across five units.

[3] BACnet_PC-NAE-1_PC-NAE-1/Programming.Air_Handling_Units.AHU-2.OA-T.PRESENT_VALUE is the full name.

[4] BACnet_PH-NAE-1_PH-NAE-1/N2_Trunk_1.AHU-20.DPR-O.PRESENT_VALUE is the full name.

		AHUs				
		DOHERTY MEC06_-DOH.AHU.A02	BACnet_PC-NAE-1_PC-NAE-1/N2-1.EN2.AHU-2	CMU/Craig Street (300 South)/Basement/AHU-3 QUIZNOS & 1 North/	CMU/SCSC Gates/Roof/ AHU-10 I/O/	MI.AHU. 3FL.011
	Vendor	Siemens	Johnson Controls	Automated Logic	Automated Logic	Automatrix
BAS tags	HW VLV	?HCO ?HCV	PH-VLV	NA	pht hw vlv	HCO
	CHW VLV	?CCO ?CCV	CLG-VLV	Cooling Valve	chw vlv	CCO
	MAT	MAT	MA-T	Mixed Air Temp	EW sup temp	NA
	OAT	OAT	OA-T [3]	OA Temp	outdoor temp	?OAC
	RAT	RAT	RA-T	Return Air Temp	return temp	?VRT
	SAT	SAT	DA-T	Supply Air Temp	Supply Air Temp	SAT
	SAT SPT	SAS	DAT-SP	Supply Air Temp Setpoint	supply setpt	SSS
	OA RH	NA	NA	OA Humidity	outdoor RH	NA
	RA RH	NA	NA	RA Humidity	return rh	?VRH
	OAD	OAD	?DPR-O[4]	Ht Whl Byp Damper	OA damper	OAD
	MAD	NA	?F_B-DPR ? DPR-O	MA Damper	recirc damper	?FBD
BAS point analysis	Total	90	33	115	138	55
	NA	3(27%)	2(18%)	1(9%)	0	2(18%)
	?	2(18%)	2(18%)	0	0	4(36%)
	✓	6(55%)	7(64%)	10(91%)	11 (100%)	5(45%)
Mapping effort	Time (mins)	11	10	13	16	10

Table 1.3: Details of mapping points in different units

Building	DOHERTY	PC	Craig Street (300 South)	Gates	MI
Vendor	Siemens	Johnson Controls	Automated Logic	Automated Logic	Automatrix
Incomplete information	25(2)	8(2)	0	0	21(4)
Missing points	17(3)	14(2)	25(1)	0	18(2)
Combined	25(5)	21(4)	25(1)	0	21(6)

Table 1.4: Number of unusable rules due to incomplete information and missing points. The number inside the parentheses represents the number of points leading to the unusable rules

We notice the inconsistency of naming does not only exist across different vendors but also for the same vendor. For example, Automated Logic uses "OA Temp" and "outdoor temp", as well as "Cooling Valve" and "chw vlv" alternatively to represent the same BAS point. This may still be understandable by facility managers and building operators; however, if we want computer programs to understand that these tags represent the same point, it is challenging to achieve without having a complete list of all possible naming conventions and the mappings.

The time spent to standardize these required points is largely due to such inconsistency. If the names of these points from different units are standardized to the same convention, a query based on this convention will be able to retrieve all the required points from multiple units. For example, a query asking for "MAT" could be used retrieve all mix air temperature sensors from many units if the points inside have been standardized to a convention where "MAT" is being used.

3. **Missing points**: four out of five AHUs do not have certain points required by APAR, which is typical for older units. The one AHU with all points available

is in a new building constructed in 2009.

Table 1.4 lists the number of rules impacted by missing points. Depending on which point is missing, the number of impacted rules varies. It is worth noting that some AHUs may not have the points required by APAR due to the specific design of the unit. For example, some AHUs do not have a heating coil and, as a result, there is no hot water valve to control the amount of hot water supplied to heat the air. For these units, the rules relying on the unavailable points cannot be applied. Another scenario is that the needed points are not being instrumented in the unit, which means additional sensing and actuating devices need to be installed to collect data in order to apply the rules.

All the difficulties mentioned above are leading to the increased cost in terms of time and efforts to retrieve the required operational information, which results in challenges of implementing FDD tools (e.g., unusable rules in the case of APAR). Specifically, the first two difficulties are due to the fact that the required information is encoded inside inconsistent or incomplete BAS tags, which prevents a clear understanding of the metadata associated with these BAS points. The third difficulty is caused by the limitation of the design of the unit and the unavailability of the hardware instrumentation. The design of the unit is inherently preventing the usage of some FDD tools as the tools are not intentionally developed for those units. The lack of hardware could be mitigated by instrumenting more sensors and actuators, which would add additional cost. Another solution to the lack of hardware is to use virtual meters to approximate the measurements from those devices. In the scope of this thesis work, the focus is to reduce the cost of retrieving the required operational information for FDD tools by tackling the first two difficulties.

1.2 Industry Challenge and Needs

Based on the case study conducted and the difficulties we came across, we identify the main industry challenge is that it is unclear whether the costs of implementing automated FDD tools in commercial buildings are less than the benefits that these tools can bring about. As a result, there are few incentives for facility managers to implement FDD tools on real buildings. For buildings with modern BASs, if the FDD tools were available as free software, then the implementation cost would be dominated by the manual mapping effort as illustrated in our case study. Given this, it is reasonable to assume that automating this mapping would tip the balance of the cost-benefit analysis.

To further illustrate this, at the beginning of the case study, when we were trying to find "MAT" in different units, we searched "MAT" directly among the BAS tags for each AHU. Unfortunately, only one of the five units produces the desired point. Since different units have their own conventions to encode the metadata, we have to spend efforts to retrieve the required points every time for a new unit. In the case study, we also observe the time spent to find the desired points increases as the number of points in that unit grows. For example, finding 11 desired points out of 138 points for one AHU in a BAS costs 16 minutes while finding the same number of points out of 55 points only costs 10 minutes, as is seen in Table 1.3.

Inconsistent naming conventions make it expensive (e.g., one day of manual investigation for one test site) to retrieve needed information to implement FDD tools [2, 17, 18, 24]. The manual cost will also increase as the number of buildings requiring FDD tools increases. As a result, it will be useful to reduce such time and efforts by finding effective and efficient ways to standardize the metadata of BAS points across buildings to a consistent naming space.

Currently, standardizing this metadata is a manual process to implement FDD tools. A simple back-of-the-envelope calculation can illustrate the cost of this manual effort. If we consider getting required operational information for one FDD tool from one building costs one day on average [17], implementing this FDD tool on all commercial buildings in the US will require 5.6 million days given there are 5.6 million commercial buildings in the US according to EIA Commercial Buildings Energy Consumption Survey (CBECS)[5]. Since the median hourly wage for a mechanical engineer (who is the typical person doing such HVAC operations) is USD $40 according to Bureau of Labor Statistics [6], the total cost of this manual process will add up to approximately $1.8 billion nationwide over all commercial buildings. The cost of implementing multiple FDD tools is even higher than that. As a result, there is a strong need to reduce the cost of retrieving required operational information from BAS, which can eventually reach a situation when the benefits can significantly outweigh the cost.

1.3 Problem Statement

This thesis deals with the problem of efficiently acquiring the consistent metadata associated with sensors and actuators (BAS points) from different building automation systems in commercial buildings, such that FDD applications can be deployed and used with minimal cost.

To illustrate and better formalize the problem, Figure 1.3 shows the information associated with one BAS point, including the observed information at the top and the consistent metadata at the bottom. The observed information from the BAS typically includes data such as time series values and metadata such as string de-

[5] http://www.eia.gov/consumption/commercial/
[6] http://www.bls.gov/oes/current/oes172141.htm

scriptors (i.e., tags), measurement units, data types, etc. As stated earlier, very often this metadata is difficult to interpret and requires experts to decode it. Additionally, different building sites tend to use distinct conventions as different vendors are involved in setting up each system.

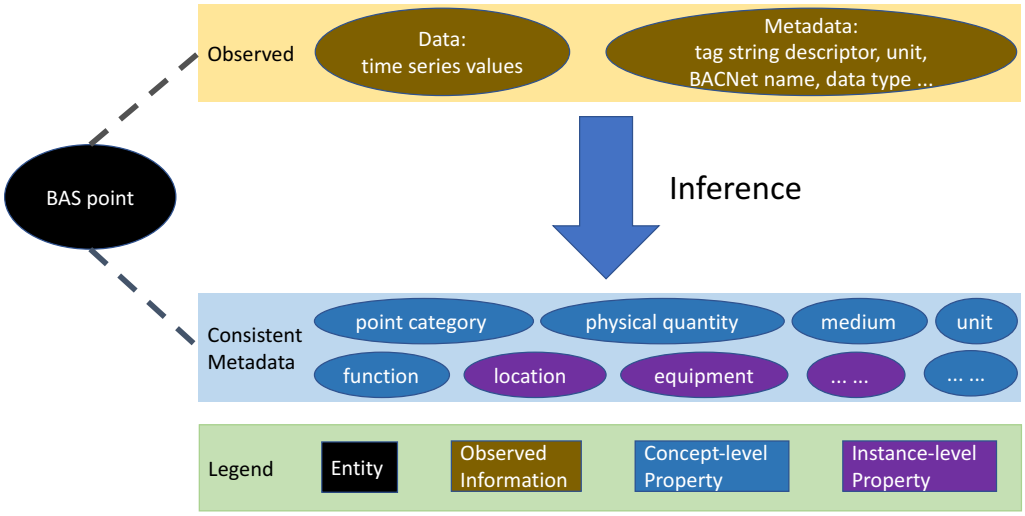


Figure 1.3: An illustration of the metadata information associated with one BAS point

The consistent metadata, as is shown at the bottom of Figure 1.3, is based on a schema that consistently describes or annotates the BAS point entity. In the figure, we divide this consistent metadata into concept-level properties and instance-level properties. Concept-level properties associated with distinct entities from different buildings can be the same as they describe the common concept in an abstract and general way. The distinct possible values of concept-level properties are finite. These properties include but are not limited to 1) the point category (it can only be sensors, set points or commands), 2) the physical quantity or phenomena the point is measuring or changing (e.g., temperature, humidity, pressure), 3) the medium the point is interacting with (e.g., water, air, steam), 4) the unit representing the magnitude of the data values (e.g., pascal, Fahrenheit), 5) the function the point is

serving(e.g., temperature of return, supply, leaving or entering medium) and others. Instance-level properties usually have their own specific representations for entities across buildings, and the distinct values of instance-level properties could be infinite such as the physical location the point resides in (site, building, floor, room, etc.), or the equipment the point is associated with such as the specific AHU, or fan coil unit (FCU).

These definitions of the data and meta-data fields have also been similarly proposed by others [28, 26, 29, 30]. Table 1.5 shows a concrete example of two BAS points from different systems describing both observed information and consistent metadata. As we can see, the consistent metadata can reduce the cost of implementing FDD applications by allowing people, and further, computer programs, to understand and interpret BAS points across buildings. Armed with this knowledge, we now proceed to review the relevant literature on the problem.

		point A	point B

Observed information	Time series data	{
		2016-01-03 9:45:20 AM: 4.75;
		2016-01-03 9:46:19 AM: 4.58;
		...
		}
			{
			2015-12-17 11:53:23 AM: 60.23;
			2015-12-17 11:54:23 AM: 60.61;
			...
			}

	Tag string descriptor	MI.AHU.3FL.011.HCO	PC-NAE-1/N2-1.EN2.AHU-2.DAT-SP
Consistent metadata	Point category	Sensor	Setpoint
	Physical quantity	Valve status	Temperature
	Medium	Water	Air
	Unit	Percentage	Fahrenheit
	Function	Heating output of the coil	Temperature of supply air
	Location	Mellon Institute	Purnell Center
	Equipment	Air handler unit - 011 on the third floor	Air handler unit - 2 in N2-1.EN2 zone

Table 1.5: A concrete example of two points in BAS where we have observed information including time series data and tag string descriptors, as well as the consistent metadata including concept-level and instance-level properties

1.4 Literature Review

The absence of consistent metadata is largely due to the lack of a common standard for people to use when the points are being defined in building systems.

Hence, many conventions, systems, and schemas have been proposed and developed [31, 2, 32, 33, 34, 18, 35, 26] to address the problem. These works attempt to either define a model to organize the metadata using a schema (focusing on the relationships between different point entities and their properties [25, 26, 36]), or suggest conventions for naming each point individually in a consistent manner (i.e., assuming that the name alone contains enough metadata information) [2, 34, 18]. Nevertheless, naming conventions are sometimes insufficient to encode complicated relationships among points and devices, and some of these schemas are oriented towards the information from the design and construction phase of the building, and cannot capture relationships and concepts needed for many applications in buildings [37, 38, 39]. To partially address limitations of existing approaches, recently the Brick [29] schema has been designed and proposed as a potential solution to manage metadata associated with entities, subsystems, and relationships among them to support portable building applications. All these efforts have made significant progress towards addressing the problem for new buildings where building stakeholders can adopt the standard schema when setting up the system. Despite that, it is still expensive to convert existing building systems to any standard manually, as is seen in [2, 17], and our case study described earlier in Section 1.1 highlights this challenge. Due to all of these limitations of pursuing metadata standardization, recent research has focused on semi-automating the process of standardizing the metadata for existing buildings by mapping BAS points to a standard schema [40, 41, 42]. The mapping is achieved through a computerized approach which learns a function that takes the information of one BAS point (e.g., the time series samples and/or the naming tags describing the point) and outputs the corresponding concepts and properties for this point in a standard schema.

These approaches, also called metadata inference approaches, can be divided into time series based, tag-based, or a combination of both. Time series based approaches utilize time series values from BAS points to learn the mapping [43, 44, 42, 45, 46]. They require the availability of historical data collected inside buildings. Tag-based approaches, on the other hand, rely on the tag names associated with BAS points [19, 24], which are determined by how vendors from different BAS companies name the points in the first place. Some researchers also use the combination of both time series data and tag names to infer metadata [40, 47, 41]. Additionally, in [48, 49], authors adopt active approaches to perturb control points to infer location and equipment connection relationships. However, unlike the other passive approaches, they require control of the system, which may only be feasible for some buildings and during specific time slots.

In terms of the metadata information being inferred, if we follow the metadata definition specified earlier in Figure 1.3, we will find many of these inference approaches focus on the concept-level properties associated with BAS points [43, 19, 40, 50, 41, 42, 45], which is also commonly referred as the "type" property.

To describe a few of these approaches, authors in [19] propose a point classification system that can assign the semantic type of the point automatically from BAS tags using the latent semantic indexing and a Naive Bayesian model. In [41] an inference approach utilizing transfer learning is adopted to learn a set of statistical classifiers of the metadata from a labeled building and adaptively integrate those classifiers to another unlabeled building to infer the sensor types. It is worth mentioning that the "type" property of the metadata can be interpreted at the different level of granularities and details. For example, researchers can distinguish "Return Air Humidity Sensor" and "Outside Air Humidity Sensor" for AHUs as two different

types or treat them as the same type like "Humidity Sensor" depending on what level of information is needed. Such a different definition of the "type" property can lead to varying performance of the metadata inference approaches. Hence, it is essential to understand what is the required "type" of BAS points that need to be inferred.

In addition to concept-level properties, there are also researchers working on inferring the instance-level properties, such as location [51, 52, 53, 44, 54], equipment associations [55, 48, 30], and other specific contextual information associated with sensors [56].

To list a few with more details, in [53] authors applied empirical mode decomposition on data from 15 environmental sensors across five rooms to find the sensors which are in the same room by analyzing correlation coefficients of intrinsic mode functions. Researchers in [44] explored how to infer the relative locations of temperature sensors with respect to each other in three rooms, by using a linear correlation and a statistical dependency measure. It is worth pointing out that both of these studies are evaluated on a limited number of rooms in a single building, which is partially due to the fact that the instance-level properties are more challenging to be inferred at scale. Recently, authors in [48] propose a new method for discovering connections between AHUs and VAV boxes from sensor data as well.

Being time series based or tag-based, focusing on concept-level or instance-level properties, we note that all these metadata inference approaches show promise to construct consistent metadata information to support building applications. However, as most of them are evaluated on a small scale, under specific building systems, the generalizability of these approaches on a large scale remains an open question. Additionally, the metadata being inferred is typically based on the information that

is available in the testbed building while the necessity of inferring this information has not been explored.

Another observation from the literature review is that most time series based approaches rely on extracting statistical properties from historical data, which we refer to data-driven models. These models are trained based on statistical features, which are another representations of the labeled BAS points using statistical quantities summarizing the time series values, and then used to make predictions for the unlabeled points. They have demonstrated to be adequate to produce metadata correctly in many cases. Nevertheless, when the model makes an incorrect prediction, it is often unintuitive to explain why it is wrong. The interpretation from statistical perspectives lacks a fundamental understanding of the underlying physics process, e.g., what are the thermodynamics driving the behaviors of each HVAC system that leads to the similar or distinct patterns in time series data.

Attempts have been made previously in [54] where authors explored the possibility of using sensor data combined with an HVAC energy estimation model to identify the exact room in which the sensor is located. This work allows for the interpretation of incorrect results from physical principles. Yet, a comprehensive explanation based on first principles for other metadata information in addition to the location information is missing, and there are significant opportunities to improve the interpretability of the statistical-based data-driven models.

1.5 Research Gaps

Our literature review is helpful to identify the following knowledge gaps:

- **Research Gap 1: the required operational information of different FDD approaches has not been identified and summarized.**

The operational information here refers to metadata associated with BAS points. Existing metadata inference approaches focus on inferring the metadata that is available in BAS. However, it has not been established whether all of the inferred metadata can be used by applications to improve operational performance. Thus, to provide operational information support for FDD applications using metadata inference approaches, there is a need to identify and summarize the required metadata for different FDD approaches.

- **Research Gap 2: the generalizability of metadata inference approaches to standardize the required operational information of a particular FDD approach has not been validated on multiple buildings.**

  Our pilot study from [42] indicates that the performance of inferring point type information can reach 75% in a single building where 20% of data are used to train the models and the remaining 80% is used for testing. Additionally, other approaches [41, 47, 40, 49] are also tested in a limited number of buildings to infer different types of BAS points. However, the effectiveness of these inference approaches to standardize the required operational information of a particular FDD approach still needs to be validated on multiple buildings, to evaluate our ability to deploy FDD applications effectively in real-world buildings in a large scale.

- **Research Gap 3: the existing metadata inference approaches based on data-driven models have limited capabilities to provide interpretability when they fail to produce the correct metadata.**

  Existing inference approaches are using data-driven models with features relying on statistical patterns and hand-crafted signatures of the data. However, as

was suggested earlier, the interpretation from statistical perspectives is limited to understand why the model fails and further provide remedies to improve the performance.

We argue that to address these three gaps, we need to develop a metadata inference framework, which can infer the required operational information for FDD approaches and generalize across multiple HVAC systems and buildings. Additionally, we need to develop a new metadata inference approach utilizing the physical models to improve our understanding of the physical process and provide interpretations when it fails.

Our vision is to enable implementing FDD tools on multiple HVAC systems and across buildings. Our envisioned framework would help facility managers and building operators to reduce the time and efforts associated with retrieving the required operational information for FDD application from BASs down to the minimum.

The envisioned outcome of this research is to go beyond deployment of a particular building application (e.g., FDD) on specific building subsystem, which may generate further impacts than FDD applications. Such a framework formalizes the metadata inference problem and provides a novel foundation for enhancing the applicability of running portable applications on multiple systems and potentially generate broader impacts in the realm of the Internet of Things.

1.6 Assumptions and Scope

There are several assumptions involved to conduct this research work to fulfill the vision of implementing FDD tools at scale. First, we assume that the design phase information is already available and that the operational information can be inferred through the information available in BAS, including both time series values and the

various metadata associated with BAS points shown in Figure 1.3. Additionally, we assume that the historical time series data collected from the BAS and used to train the metadata inference algorithms are not dominated by faulty conditions. That is, though we cannot directly verify this, we assume that the data used for training the algorithms are not so corrupted by faults that it makes the conditional distribution of the data given its sensor type useless for the inference procedure.

The scope of this research will be limited to FDD on secondary HVAC systems in commercial buildings, which are about airside distribution systems including AHUs, FCUs, and terminal boxes (VAV and constant air volume (CAV) boxes). The reason for setting this scope is because more than 60% of the energy wasted by faults is caused by duct leakage, stuck fan, and other similar issues related to components in the secondary HVAC system [3], i.e., the air side of HVAC system including VAV air conditioning systems. From now on, we will use FDD approaches to explicitly represent FDD approaches in secondary HVAC systems on commercial buildings. Work on the primary HVAC systems including the waterside system (e.g., chillers and boilers) will be left as the extended future work.

Additionally, we will also limit the scope to focus on inferring the concept-level properties associated with BAS points using time series based metadata inference approaches. The inference of instance-level properties along with other metadata inference approaches could be studied by extending the framework proposed in this thesis.

1.7 Research Questions

To address the research gaps identified, we propose three research questions (RQ).

1. **What is the specific operational information required by FDD approaches for secondary HVAC systems found in existing literature?**

   The specific operational information refers to metadata associated with BAS points, including both concept-level and instance-level properties.The purpose of this question is to understand this metadata such that it can be inferred using metadata inference approaches. We select and review 110 academic publications about FDD approaches, to identify the required operational information for each of them. This information is summarized and analyzed to reveal the commonly required information, as well as other relevant patterns (e.g., which is the most popular HVAC subsystem that FDD approaches have been applied to).

2. **How is the performance of existing metadata inference approaches affected by changes in building characteristics, weather conditions, building usage patterns, and geographical locations?**

   As is stated earlier, the proposed framework utilizing metadata inference approaches should not be tested only on a single building; instead, the effectiveness should be validated on multiple buildings with different building characteristics to understand whether it can generalize well. The generalizability can be defined as the performance of the model when we vary the weather conditions and the geographical locations of buildings being tested. This performance can be quantified with metrics like the accuracy, the F_1 score, etc.

3. **What is an approach that can be developed to complement existing data-driven models by providing physical interpretations in the case of incorrect metadata being inferred?**

Existing approaches are based on data-driven models utilizing statistical patterns lacking the ability to capture the intrinsic physical dynamics of the HVAC systems. We seek to develop a new approach based on the physical models of HVAC systems to understand the behavior of each component and provide physical interpretations for the incorrect inference outcome, which complements existing approaches.

1.8 Document Organization

The thesis is organized as follows. In Chapter 1 we introduce the problem to be solved, and we define the scope of the research including three research questions. To answer each of the research questions, in Chapter 2, we identify the operational information from a list of FDD approaches. Then in Chapter 3 we present the metadata inference framework and evaluate it on hundreds of buildings to study the generalizability. We then have a small digression in Chapter 4 to explore the pure data-driven approach using a convolutional neural network to tackle the metadata inference problem. Lastly, we improve our understanding of metadata inference approaches leveraging the physical models of the AHU in Chapter 5. We finally present Chapter 6 to conclude the thesis and discuss the future work that can follow.

Chapter 2

Identification of Required Operational Information from FDD Approaches

Over the past thirty years, hundreds of FDD approaches have been developed for HVAC systems in large commercial buildings to bring about benefits including energy savings, increased operating efficiency, reduction of maintenance cost, improved occupants' comfort and productivity, etc. These approaches leverage data collected in buildings to detect any system malfunctions and provide diagnosis capabilities. One piece of the required input information for these approaches is the operational information, which is also referred as the concept-level and instance-level properties associated with BAS points.

For different FDD approaches, this required operational information, also known as metadata, has not been well summarized and documented, which prevents people from applying metadata inference approaches effectively to produce this desired

metadata. Hence, it is necessary to identify and summarize what the required BAS points along with the metadata associated for each of FDD approaches are. This metadata is also referred to as the "type" of BAS points. For example, a BAS point of type `supply air temperature sensor` from an AHU has point category "sensor", measures the physical quantity "temperature" through the medium "air", serving the function of "tracking the temperature of supply air in an AHU". The unit (Celsius or Fahrenheit) can be further derived from actual data values. This representation encodes the necessary concept-level properties. The instance-level properties are typically less relevant, as FDD approaches are not designed to be used only in specific equipment in a particular building. In the following contents of this chapter, we will use point types to refer to the concept-level properties, which is also the operational information we will identify.

The importance of identifying the information requirements for HVAC applications has previously also been addressed in [57] where the author proposes an approach to identify a general set of information requirements for performance analysis and improvement of HVAC systems. The author further provides a very detailed classification of these general information requirements in terms of HVAC components, HVAC subsystems, and building design information. The novelty of our work consists of focusing on the identification of more specific operational information (types of BAS points) required by each of FDD approaches (e.g., mixed air temperature sensor, supply air temperature sensor, return air temperature sensor from an AHU), as previous work focused on summarizing the general set of information requirements (e.g., type, location, medium of transporting components).

In this chapter, we first select a list of FDD approaches described in academic publications that include BAS points as the input. Then we identify and summa-

rize the types of BAS points for each of these selected approaches. This identified operational information will be analyzed to understand:

1. What are the most commonly required types of BAS points for FDD approaches?

   The answer to this question provides guidance regarding what BAS points and associated metadata should be inferred using metadata inference approaches. As a result, these inference approaches can be effectively used to provide the operational information support and facilitate implementing FDD applications on multiple HVAC systems.

2. What are the characteristics of the existing FDD approaches in terms of their type and targeted system?

   The answer to this question provides sights into the development status of FDD approaches in secondary HVAC systems, which allows FDD research community to know which FDD approach type (including qualitative, quantitative and data-driven models defined in [9]) is most popular, what targeted systems have been focused on most. The existence of this information associated with each FDD approach may also provide a starting point for formalizing FDD approaches and their information flow.

3. What is the coverage of identified point types in existing buildings and what are the missing points in existing buildings to deploy FDD applications?

   This helps to understand the feasibility of implementing FDD at scale. Additionally, it also guides engineers to be aware of what hardware should be instrumented if specific FDD applications are desired when initially set up the building.

2.1 Selection of FDD Approaches

Common FDD approaches usually involve comparing the actual performance of the system to a reference [13]. A high deviation between the reference and the actual performance, which is either set heuristically based on experts' domain knowledge or decided with certain statistical significance from data (e.g., the value that is three standard deviation out of range of historical values), indicates the existence of a fault. One example we have encountered in Chapter 1 is a rule-based FDD approach from [15] named APAR. The required operational information is shown in Table 1.1 concerning 11 types of BAS points. In this section, our goal is to find a list of FDD approaches in addition to APAR, which is then used to extract the required BAS points.

Given the existence of many FDD approaches described in academic publications such as journals, conference proceedings, books, theses and technical reports, examining all possible FDD approaches and identifying the BAS points needed by them would be unrealistic as it is non-trivial to find all references describing the FDD approaches, and it would take a significant amount of time to identify required point types from all of them. Moreover, very likely, the distribution of point types might converge after examining a certain number of publications. Hence, instead of selecting FDD approaches out of all possible publications[1], we start with a pool of references which are cited in seven publications [9, 10, 58, 59, 60, 61, 57]. These seven publications all have a long reference list, and they summarize and discuss a large number of FDD approaches.

[1]Ideally, we could build a directed graph made of publications with each node pointing to the publication that cites itself. By running a depth-first search querying all the nodes with a particular condition, we could potentially find all relevant FDD approaches. However, it is rarely possible to build the graph in the first place as we do not have full access to a massive publication database tracking the citation information like Google Scholar.

After removing duplicate references, the total number of unique references in our pool is 745. As we are only interested in FDD approaches that allow us to extract a specific set of BAS points, we select these relevant FDD approaches by considering the following two major criteria:

1. **Is it applied to secondary HVAC systems?**

   As the research scope is about FDD approaches on secondary HVAC systems in commercial buildings, we focus on the air-side systems such as AHUs, VAVs, FCUs, etc. References including FDD approaches applied only to primary HVAC components (e.g., chillers, boilers) are thus removed. There are scenarios on which some FDD approaches have been applied to both primary and secondary HVAC systems, in which case we keep the reference. Additionally, publications which are targeting residential buildings (e.g., FDD in refrigerators, or residential heat pumps) are also filtered out.

   This criterion reduces the reference count significantly from 745 down to 181, which filters out references such as 1) reviews/surveys/handbooks; 2) papers about FDD which are applied to chemical process, energy consumption of thermal plant, and others instead of HVAC systems; 3) papers about HVAC systems that focus on control optimization, performance improvement, HVAC component design, energy conservation, and others instead of fault detection and diagnosis.

2. **Are the required BAS points distinguishable in this publication?**

   We consider the publications which include an FDD approach that can be implemented. In other words, the FDD approach has to specify a set of BAS points as the input. Using this constraint, we remove some software FDD tools

(e.g., DABO [62, 63]) which we do not have access to the internal documentation or source code regarding how the approaches are being implemented to replicate them. Additionally, we also remove the publication discussing the general FDD methodology without specifying details, for example, [64] describes a hierarchical rule-based integration methodology which assumes the faults from sub-components have been detected.

This criterion eventually leaves us 110 publications describing 110 selected FDD approaches which we will use in the next section to extract the required operational information regarding BAS points.

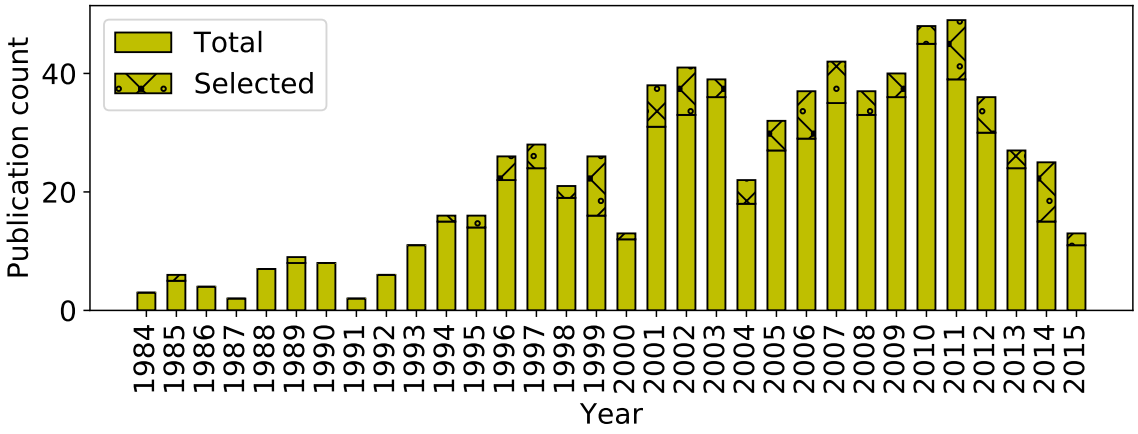


Figure 2.1: Counts of total and selected publications from 1984 to 2015

Figure 2.1 shows the publication count from 1984 to 2015. The vertical yellow bar represents the total count of reference for that year while the hatched section at the top represents the relevant references connected to the FDD approaches we select. As we can see, the overall trend of the research in the field of FDD and HVAC is increasing with a relative decrease in recent years. The drop is because only a few publications after 2015 are included in the pool of references as we extract

those references from (old) review papers and there is a delay between submission and publishing.

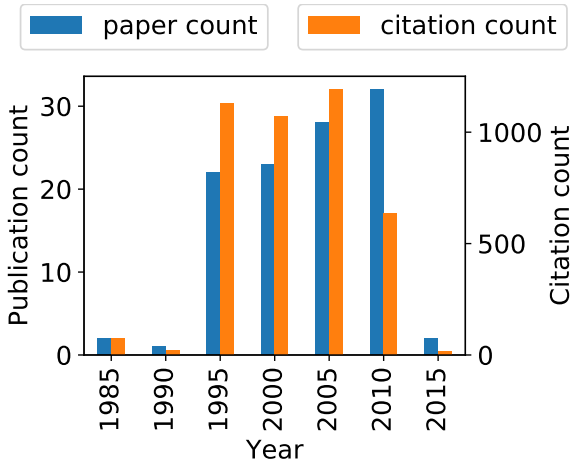


Figure 2.2: Publication counts and citation counts every five years

We further extract the citation count of these 110 selected publications from Google Scholar. Figure 2.2 displays the paper count and citation count every five years. We can see the same trend that more publications about FDD approaches are published over the year with a decrease in recent years. It is natural to see more citations for the old publications and fewer citations for the new publications. Another interesting finding is that the top 3 most cited publications are [65] in 2004, [15] in 2006 and [14] in 2001 with 172, 131, 122 citation counts each among these 110 FDD approaches. The first one is a principal component analysis (PCA) based approach to diagnose the faults in AHU, and the latter two are both using rule-based FDD named APAR.

2.2 Identification of the Operational Information

Once these 110 publications are selected, we need to identify the operational information for each of them. Despite the existence of the approaches like Informa-

tion Delivery Manual (IDM) [66] developed by BuildingSmart to unite information from different business processes through the Architecture, Engineering, Construction and Facility Management (AEC/FM) project life cycle, or affinity diagramming method [67] to discover common themes and issues among different work practices to extract information items, many of the middle processes used in these approaches (e.g., building process maps) still rely on human subjective interpretations depending on how the FDD approach is described in the publication and how the reader understands the methodology. Perhaps one future direction to tackle such challenge is formalizing FDD approaches and their information flows. In [68], the authors specify FDD approaches with ontologies including information from self-description, requirements, and configurations. Nevertheless, the process of converting each FDD approach to this specification is still subjective, and the specification approach lacks a formal description. Thus, presently, the task of extracting required operational information for each FDD approach still requires manual efforts and subjective decisions.

For some FDD approaches, the required BAS points are explicitly enumerated, listed in tables or described in the text, as is seen in Figure 2.3. The most common way of describing the required information is by texts as is seen in Figure 2.3b and Figure 2.3c. For the FDD approaches which specify the required BAS points clearly, different people will identify the same set of consistent BAS points as long as the person has sufficient background knowledge to understand FDD approaches.

However, there are some other FDD approaches which describe the methodology and the implementation steps without enough detail. We are aware that in this case subjective decisions could be made regarding identifying required BAS points. For example, in [72], the authors specified θ_S, "air temperature measured by the sensor"

APAR uses the following occupancy information, setpoint values, sensor measurements, and control signals:

- Occupancy status;
- Supply air temperature set point;
- Supply air temperature;
- Return air temperature;
- Mixed air temperature;
- Outdoor air temperature;
- Cooling coil valve control signal;
- Heating coil valve control signal;
- Mixing box dampers control signal;
- Return air relative humidity (for enthalpy-based economizers only);
- Outdoor air relative humidity (for enthalpy-based economizers only).

(a) Points are listed separately [15]

PCA models to make correlations closer. The PCA model based on the heat balance involved nine variables: M_{fre}, M_{sup}, M_{rtn}, T_{fre}, T_{sup}, T_{rtn}, h_{fre}, h_{rtn}, and $C_{val,\ w}$, which constructed a nine-dimensional measurement space. This model could detect faults in eight sensors

(b) Points are embeded in texts [69]

system. In this paper, seven relative sensors are selected for this research objective. There are respectively outdoor air flow sensor, total supply air flow sensor, return air flow sensor, outdoor air temperature sensor, VAV supply air temperature and CAV(constant air volume conditioner) supply air temperature sensor. Data derived from those seven

(c) Points are embeded in texts [70]

Table 1
Typical sensors list of AHU operation.

Outside air temperature	Mixing box damper position signal	Supply fan total power meter
Mix air temperature	Heating coil valve position signal	Return fan total power meter
Supply air temperature	Cooling coil valve position signal	Supply fan speed signal
Supply air duct static pressure	Supply airflow rate	Return fan speed signal

(d) Points are in a table [71]

Figure 2.3: Examples of how input BAS points are presented in different FDD publications

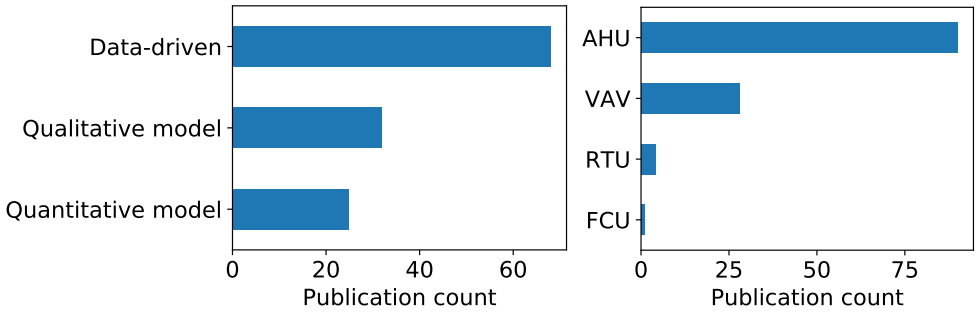
in an AHU as one of the inputs to the model without providing the particular function of that sensor. Based on the context, we infer it to be "supply air temperature sensor" from an AHU. Such decisions could be subjective and different people may draw different conclusions. To mitigate the issue, we will: 1) open-source all the results of the identified operational information where we will display the identified BAS points for each of 110 FDD approaches; 2) mark the FDD approaches without enough implementation details and conduct the analysis with and without these approaches.

In addition to the required inputs for FDD approaches, we also extract the infor-

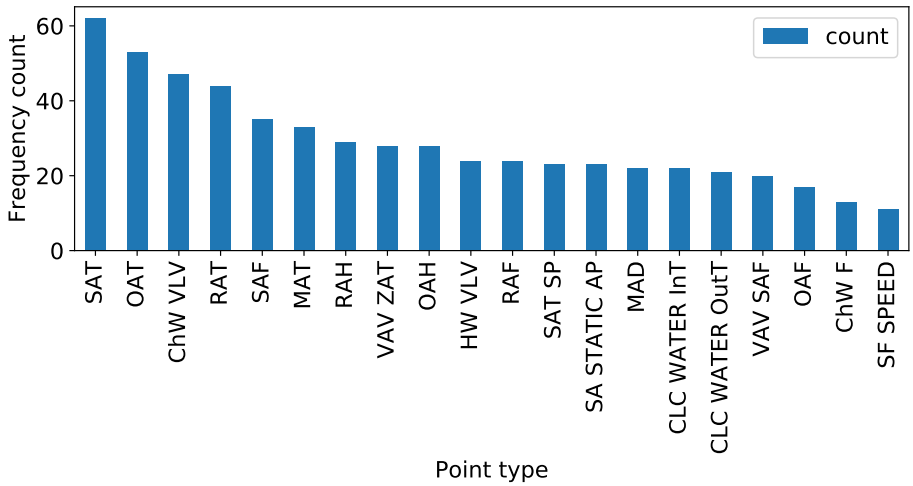
mation including the types of FDD approaches and the targeted systems. The type is defined following the definition in [9] including qualitative, quantitative and data-driven models. The targeted systems include AHU, VAV, FCU, packaged rooftop unit (RTU), and others which are related to the air-side of HVAC systems. It is worth mentioning that some approaches may use a hybrid approach including both qualitative and data-driven models targeting multiple sub-systems like both AHU and VAV. In those scenarios, we will associate both types and targeted systems with this approach.

After examining each FDD approach and identify the required BAS points, we mark 24 out of 110 approaches without explicitly specifying the required inputs. We plot the publication count for each approach type and target system. For the point type, we show the 20 most frequent ones based on a total of 110 publications and a reduced set of 86 publications after excluding FDD approaches without clear specifications. Figure 2.4 demonstrates the frequency counts for different classification methods. It is worth noting that one publication could be counted multiple times if it includes multiple types of approaches or targeted systems. As we can see, about 60% of the FDD approaches in our study are developed using data-driven models. Also, most of FDD approaches developed for secondary HVAC systems are designed for AHU equipment.

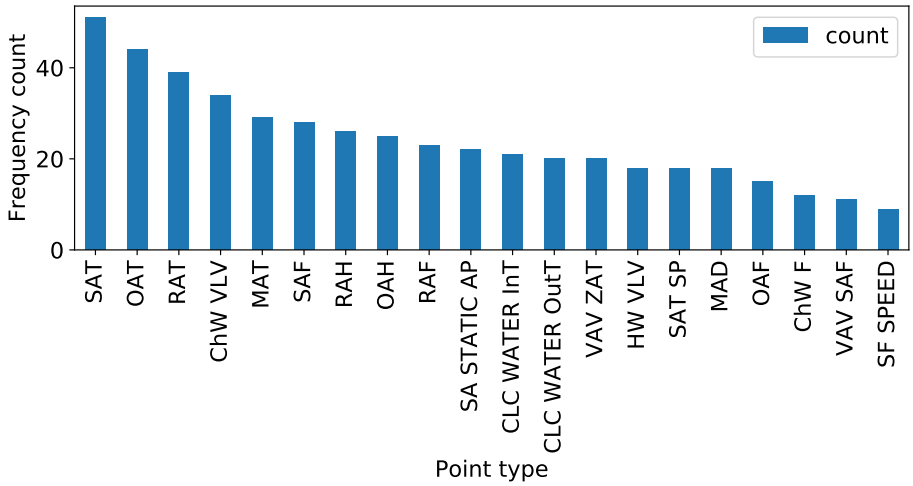
Regarding the BAS point types, we do not observe much difference when excluding 24 FDD approaches we are uncertain about. The top 20 frequent types from both sets are the same with the only difference being the orders of their most frequent types are not matched. The most widely used point types in FDD approaches are points in AHUs which include “supply air temperature”, “outside air temperature”, etc. A mapping between the acronyms and the full descriptions of



(a) By approach type (b) By targeted system



(c) Publication counts of 20 most frequent point types



(d) Publication counts of 20 most frequent point types after excluding uncertain FDD approaches

Figure 2.4: Publication counts by different classification methods

the BAS points can be seen in Table A.1, A.2, A.3, which include points for AHU, terminal box, and RTU, respectively. As the points required by FDD approaches in FCU only include the fan power, we will only mention it here and will not list it separately to a table in the appendix.

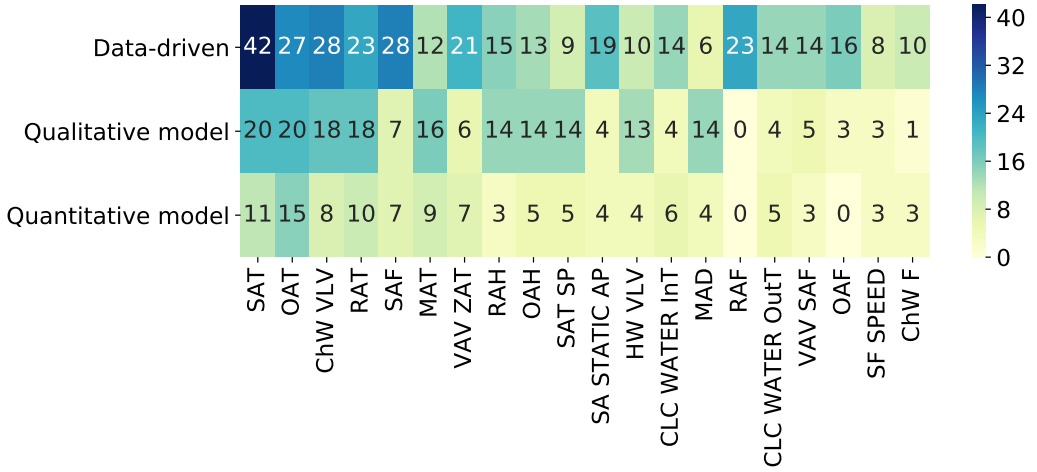


Figure 2.5: Publication counts by sensor type and approach type

We also explore how each sensor type is preferred for each approach type and target system. Since excluding 24 FDD approaches we are uncertain about rarely affect the distribution of frequent types, we only show the results including all 110 publications in the following plots to have more statistical significance. Figure 2.5 displays a 2D heat map showing the publication counts by sensor type and approach type. We can see how frequent the temperature related measurements are needed in all approach types.

Figure 2.6 shows a 2D heat map exhibiting the publication counts by sensor type and target system. Given the number of BAS points from RTU and FCU is small, we only show AHU and VAV here. The reason we still need "VAV ZAT" in AHU-targeted approaches is that we double count the required points for 13 FDD approaches which are applied to both AHU and VAV systems. Additionally,

there are some AHU-targeted approaches which do rely on the zone air temperature measurements from the VAV box that the AHU is supplying air to and some VAV-targeted approaches also require sensors from the AHU.

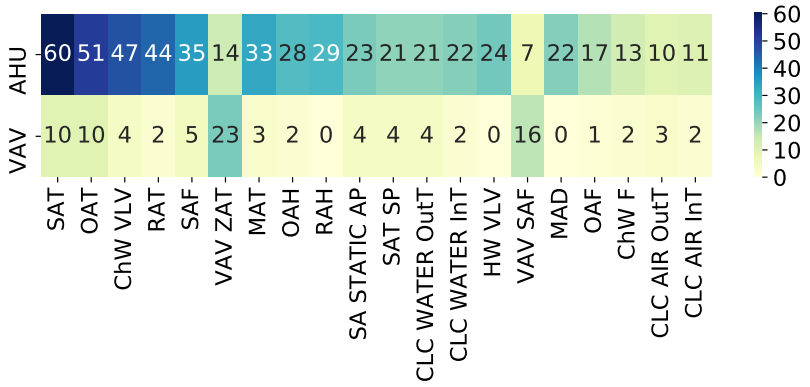


Figure 2.6: Publication counts by sensor type and targeted system

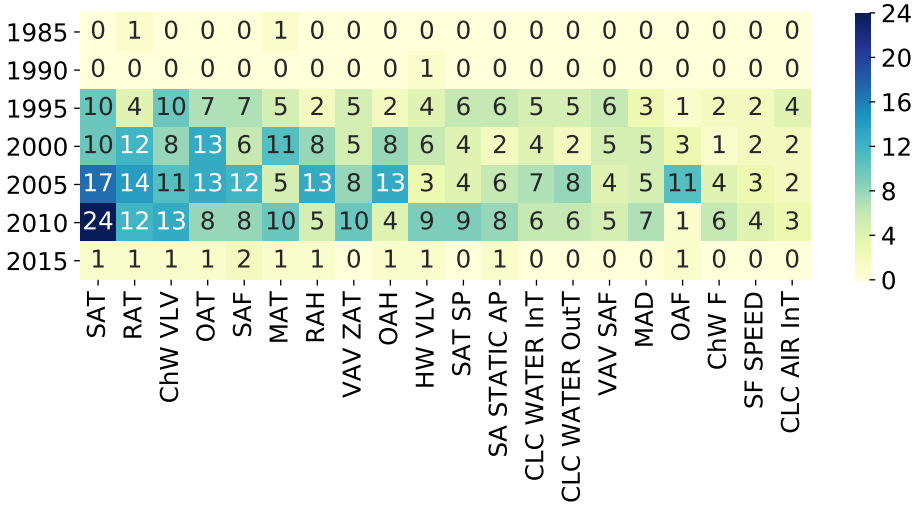


Figure 2.7: Publication counts by sensor type and every five years

Figure 2.7 shows a 2D heat map displaying the publication counts by sensor type and time (every five years). If we ignore the row after the year 2015, we can see the overall trend of the publication counts regarding most frequent used BAS points is increasing.

A total of 102 different BAS points required by FDD approaches are identified from 110 publications, which are then classified into three groups belonging to AHU, terminal box, and RTU. The details can be seen in Table A.1, A.2, A.3. We observe that as more FDD approaches are reviewed, the number of new BAS points that are identified from each additional approach becomes smaller. This is validated by tracking the cumulative count of identified BAS points when we review more FDD approaches. Since the sequence of the reviewed FDD approaches will impact the trend of the cumulative count, we generate K random sequences representing K possible ways of reviewing FDD approaches in order. For each sequence i, we track the cumulative count of identified BAS points $\mathbf{p}^{(i)} = \{p_1^{(i)}, p_2^{(i)}, \cdots, p_{110}^{(i)}\}$ when the number of reviewed FDD approaches increases from 1 to 110. Given K sequences, we calculate the average case $\bar{\mathbf{p}} = \frac{1}{K}\sum_{i=1}^{K}\mathbf{p}^{(i)}$, the upper case defined as $\bar{\mathbf{p}} + 3\boldsymbol{\sigma}$, and the lower case defined as $\bar{\mathbf{p}} - 3\boldsymbol{\sigma}$, where $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \cdots, \sigma_{110}\}$, $\sigma_j^2 = \frac{1}{K}\sum_i^K (p_j^{(i)} - \bar{p_j})^2$, $\bar{p_j} = \frac{\sum_i^K p_j^{(i)}}{K}$. To have enough statistical significance, we choose $K = 10000$. The resulting three curves of the cumulative count for each case can be seen in Figure 2.8. The two black straight line mark the position of 70% and 90% of the cumulative counts of BAS points. In order to cover 70% of BAS points, we need to review 38% / 18% / 62% of publications under each case. The bound indicates that 99.7% of time we are able to cover more than 70% of BAS points by reviewing 18% to 62% of publications. Similarly, if we extend the coverage to 90%, the number will change to 72% / 40% / 91% of publications respectively. This conclusion is similar to what is claimed in [57].

We also briefly examine two of the existing schemas, Brick [29] and Haystack [26], to verify if they can cover all the identified BAS points. We find that neither of them can express all the information items without first extending the schema. On

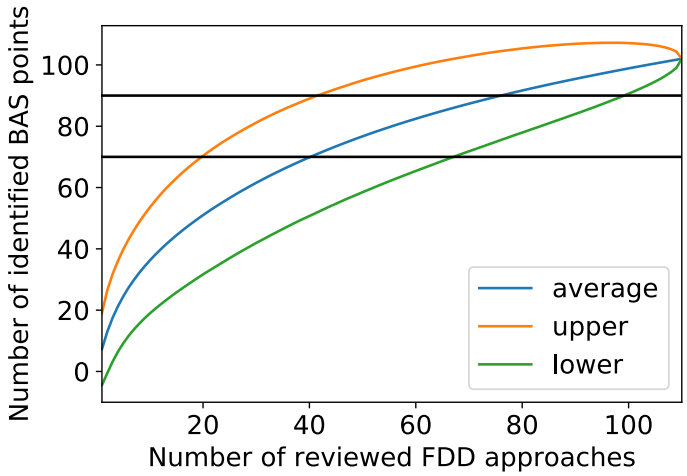


Figure 2.8: Cumulative counts of identified BAS points using random order selection of reviewed FDD approaches

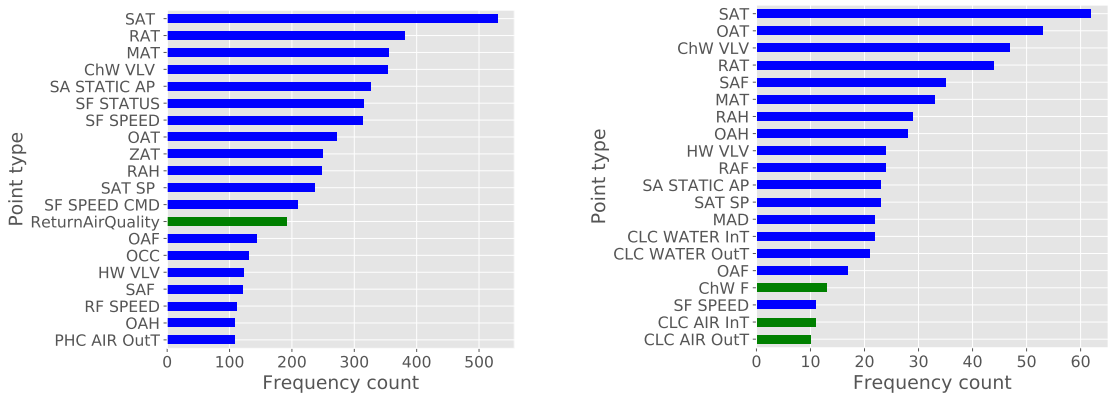
the one hand, Brick does not have RTU related points, such as the air inlet and outlet temperature of an evaporator inside an RTU. On the other hand, Haystack does not capture power related measurements, such as pump power, fan power, etc. This might suggest the need to develop an FDD-specific schema that will contain all the required information for FDD applications in future, perhaps by extending one of the existing ones.

2.3 Coverage of Identified Information in Existing Buildings

It is not uncommon for the operational information required by FDD approaches to be different from the BAS points available in buildings. In this section, we analyze points in AHUs from existing buildings to understand how often these commonly required BAS points are covered.

We have access to a dataset of 6145 BAS points from 614 AHUs inside 421 buildings. These buildings are grouped into 35 different sites across the US. The

details of this dataset are further explained in Section 3.3. We use this dataset to understand how many AHUs have all the required BAS points for a particular FDD approach. For this purpose, we select 69 FDD approaches applied to AHUs with an explicit set of BAS points required. To understand the distribution of points, we plot the top 20 frequency point types from both FDD approaches and existing buildings in Figure 2.9.



(a) From BAS, the green one represents the point that is found in BAS but not required by FDD approaches

(b) From FDD approaches, the green ones represent those required by FDD but not found in BAS

Figure 2.9: Top 20 frequent points from BAS and FDD approaches

As we can see, the overall distributions of frequent types existing in BAS and required by FDD approach are similar where they share 12 out of 20 types. There are point types which are in BAS but not in FDD (e.g., "ReturnAirQuality") and there are also points in FDD approaches but not in BAS (e.g., "ChW F", "CLC AIR InT" and "CLC AIR OutT"). Among 75 types of BAS points in AHUs, only 36 (48%) types can be found in identified required BAS points for FDD approaches; however, they cover 86% of total points in AHUs.

For each FDD approach, we can impose a hard constraint by counting the number of AHUs that contain all the required points of this approach. The top five FDD

approaches that are more easily applied to AHUs are demonstrated in the Table 2.1.

year	title	AHU count	required inputs	citation count
1985	An innovation-based methodology for HVAC system fault detection	315	MAT, OAT, RAT	50
2012	A rule augmented statistical method for air-conditioning system fault detection and diagnostics	188	MAT, ChW VLV, SAT, SA STATIC AP, VAV RH VLV, VAV DP, VAV ZAT, VAV SAF	9
2001	EMCS and time series energy data analysis in a large government office building	152	MAT, RAT, OAT, SAT, SAT SP	13
2005	Transient pattern analysis for fault detection and diagnosis of HVAC systems	77	OAT, SA STATIC AP, OAD, SAT, VAV ZAT	45
2006	Automated Fault Detection and Diagnosis for an Air Handling Unit Based on a GA-Trained RBF Network	36	SA STATIC AP, SAF, SAT, ChW VLV, SF SPEED, RF SPEED	2

Table 2.1: Top five FDD approaches that can be applied to most AHUs

Among all 69 FDD approaches applied to AHUs, there are 52 of them which cannot be applied to any of 614 AHUs. This is mainly due to the fact that a significant number of AHUs do not have all the required points of FDD approaches, such as supply air flow rates, outside air flow rates and mixed air dampers, which are commonly required by FDD approaches but not available in BASs. Table 2.2 shows the top 10 missing points in AHUs that lead to this problem. The problem can be alleviated by instrumenting these missing points in AHUs or using virtual sensors to approximate them.

2.4 Conclusion

In this chapter, we review and select 110 publications out of 745 references. The selected papers include FDD approaches applied to secondary HVAC systems with distinguishable BAS points as their inputs. We identify a total of 102 different

Acronym	Description
SAF	supply air flow rate
CLC WATER InT	cooling coil inlet water temperature
RAF	return air flow rate
CLC WATER OutT	cooling coil outlet water temperature
MAD	mixed air damper position
RAH	return air humidity
RAT	return air temperature
ChW VLV	chilled water valve position
HW VLV	hot water valvle position
ChW F	chilled water flow rate

Table 2.2: Top 10 missing points in AHUs

BAS points required by selected FDD approaches. These identified BAS points are further summarized and analyzed to obtain the most common points and compared against the points in real buildings. We find that:

- The BAS points required by more than 30% of FDD approaches include six sensors monitoring supply air temperature, outside air temperature, chilled water valve position, return air temperature, supply air flow rate, and mixed air temperature, which are all instrumented inside AHUs. This suggests that the metadata associated with these six sensors should be inferred using metadata inference approaches such that more FDD applications can be easily implemented.

- Data-driven models are more prevalent which occupies 62% of total approaches reviewed (68 out of 110), and 82% of developed FDD approaches (90 out of 110) can be applied to AHUs. Given the number of FDD approaches developed based on data-driven models and targeting AHUs, there is a need to facilitate converting applications that can be used in real-world buildings. Meanwhile, it also suggests the necessity of developing FDD approaches applied to other

HVAC subsystems in addition to AHUs.

- The overall distributions of frequent point types existing in BASs and required by FDD approaches are similar where they share the same 12 out of 20 types. Given BAS points available in an AHU, we check which FDD approaches can be applied to this unit and find one FDD approach can be deployed to 315 out of 614 AHUs. Since some AHUs do not have all the required BAS points for FDD approaches, we track the frequent missing points in AHUs with top three being supply air flow rate, cooling coil inlet water temperature and return air flow rate. This guides building operators to instrument additional sensors if more FDD applications are desired to be implemented.

These findings, though useful, do not fully address the practical challenges encountered by facility managers wishing to implement FDD approaches on their systems. For instance, it would be more meaningful to extend the analysis to include information directly related to the energy saving potential of the different FDD algorithms so that decisions about which algorithm to implement given the available information can be made more objectively (and based on measures that are more meaningful). Our analysis, however, is limited to extracting the required information items for different FDD approaches without making any specific consideration of how this information will be used. Showing the connecting between these identified requirements and energy savings or other benefits will be left as one potential future avenue of research.

To sum up, this chapter identifies the required BAS points for different FDD approaches and guides the metadata inference approach to produce the required metadata for FDD applications, which will be the focus of the following chapters.

Chapter 3

A Metadata Inference Framework Applied to Hundreds of Buildings

The lack of consistent metadata associated with BAS points across buildings is one of the main reasons preventing the deployment of building applications, which motivates the development of many metadata inference approaches [43, 44, 42, 45, 46], as is discussed earlier in Section 1.4. These approaches have shown the potential of standardizing metadata and further facilitating the deployment of portable FDD applications. Nevertheless, each FDD application for which we use metadata inference approaches would have different sets of required BAS points, and each inference approach would obtain those points with different performance. For example, if we were interested in deploying an FDD algorithm in local zone VAV boxes that required access to zone-level temperature setpoints and temperature measurements, it is not clear which of the existing metadata inference approaches would be best suited to support this application. Moreover, studies to date have been preliminary, and most of the approaches have been evaluated only on a small scale (typically on

This chapter is partially based on [73].

two or three buildings). The generalizability of these approaches on a large scale remains to be investigated. Additionally, the amount of human work required to configure and run each inference approach in order to achieve its best performance in a given application also varies. Hence, there is a need to better understand the trade-offs of these choices.

In this chapter, to shed light on these choices and improve our understanding of the limitations, we propose a metadata inference framework that provides operational information for FDD applications. Using the framework, we evaluate six metadata inference approaches on more than 400 buildings to infer the BAS points required by a particular FDD application (APAR), which has been used by two of the top three cited publications on FDD identified in Chapter 2. Since there is considerable consistency in the tags used in our dataset (given that they come from a single vendor), we limit our scope to time series based approaches. Specifically, we infer 12 types[1] of BAS points collected from 614 AHUs from 421 buildings across 35 different sites across the US. By applying this framework to evaluate six different inference approaches on hundreds of buildings, we want to find out how will the performance of existing metadata inference approaches change when we vary the building characteristic under different weather conditions and geographical locations. Specifically, we want to answer the following questions:

1. Is there one metadata inference approach that generally works well on different building sites?

2. Is the information available from a subset of buildings rich enough to represent the distribution of another group of buildings?

[1]Notice the type of BAS points in this context encodes the concept-level properties of the metadata.

3. How will the performance of inference approaches be affected when we vary the building characteristics, weather conditions, and geographical locations?

3.1 Framework

We first introduce the metadata inference framework to provide operational information support for FDD approaches shown in Figure 3.1.

Notice the goal of the framework is to provide the operational information, i.e., BAS points with consistent metadata, such that FDD applications can be easily implemented in multiple buildings. Therefore, the first step is to specify which FDD approach to be implemented and identify the required BAS points for that approach. This step has been done in Chapter 2.

Then we need to extract these required BAS points from existing buildings. As it was shown earlier in Figure 1.3, we typically have access to the observed information associated with BAS points, which includes time series values and very often obscure metadata descriptions. Based on the information of identified required BAS points for FDD approaches, we can label a small portion of the observed BAS points with consistent metadata following a unified standard or schema. The choice of the standard or schema is less relevant as long as it can encode all the required metadata and the metadata representation is consistent. Our experiments were carried out using the Brick metadata schema. This small portion of labeled BAS points will be used as the training data to build the model. The model is then used to predict the consistent metadata for the testing data, which are the unlabeled BAS points in buildings.

It is worth pointing out that by using different strategies to produce this small portion of training data and the remaining testing data, the performance of the inference approaches will also vary. A good strategy should satisfy the specific use

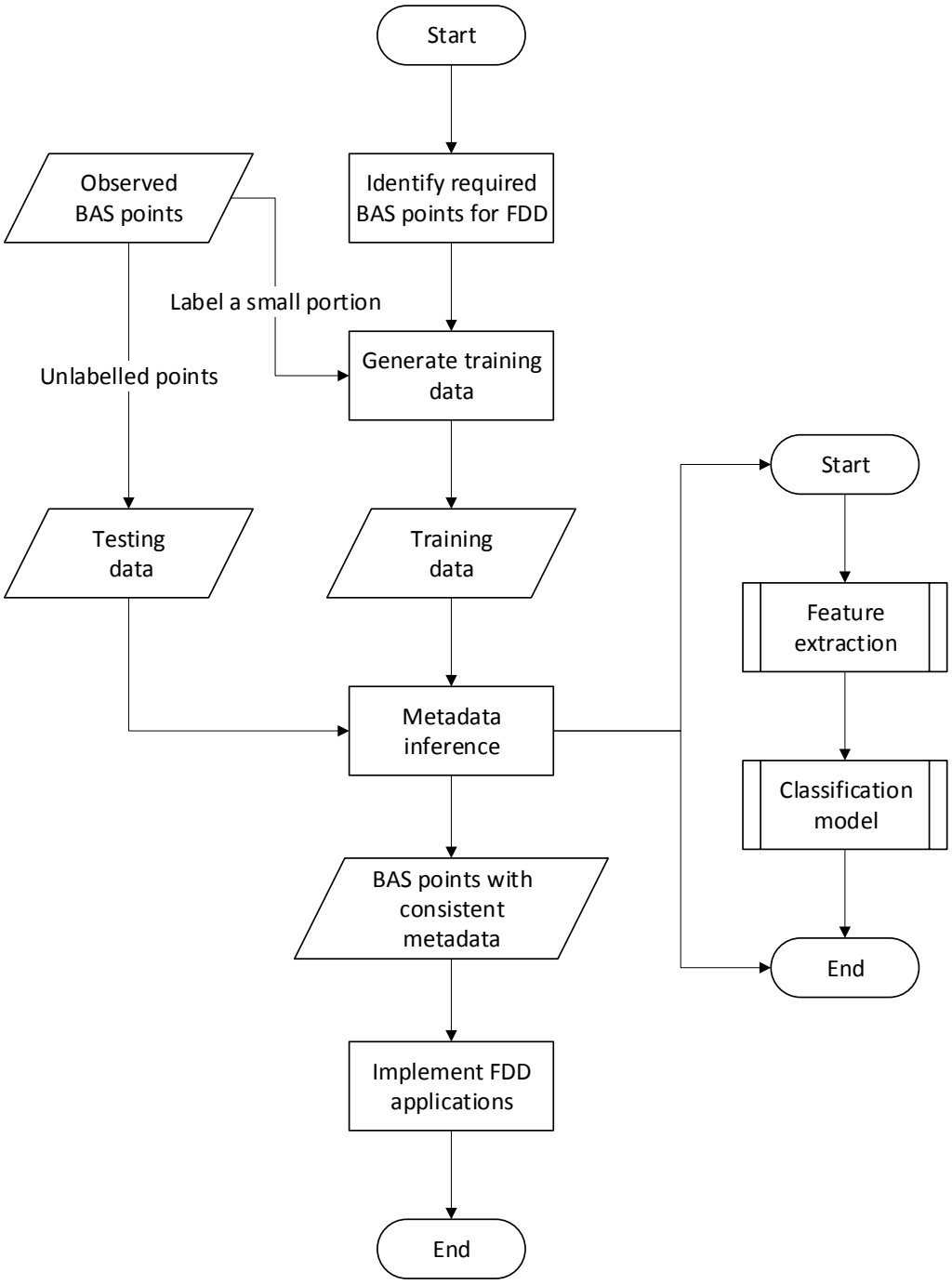


Figure 3.1: A metadata inference framework to provide operational information for FDD applications

case for different people and should try to minimize the amount of manual labeling efforts and to maintain an acceptable level of performance such that it can be realistic enough to be used in practice. For example, for a building manager with hundreds of buildings where BAS points need to be labeled, she or he may prefer being able to train a model on some labeled buildings and use the model to produce the consistent metadata for the rest buildings, instead of labeling some data from a new building every time the consistent metadata needs to be produced.

Once the training data and testing data are generated from the observed BAS points, we feed training data as inputs to the metadata inference approaches to train different models. These models can produce the labels (consistent metadata) for the remaining BAS points, i.e., testing data. The consistent metadata for all BAS points is eventually integrated and used to implement corresponding FDD applications.

One of the key components is the metadata inference process, which involves two major steps: feature extraction and classification. These two steps are what typically distinguish the different metadata inference approaches in the literature. We will discuss this process in more details the next section.

3.2 Methodology

As stated previously, the goal of this chapter is to address the generalizability of metadata inference approaches. Hence, we evaluate six time series based approaches [74, 42, 41, 47, 40, 49] on more than 400 buildings. Five of these approaches were selected based on a literature review that we conducted to find time series based metadata inference approaches applied to building automation data to infer sensor types, and they represent the totality of the publications we found meeting that criteria. However, we realize that there may be other approaches that exist,

and many more will be developed later, so we leave it to other researchers to extend the evaluation work. The sixth approach was taken from the database community where it is applied to the problem of schema matching [75], which shares many similarities with the metadata inference problem in the building community. Mapping inconsistent metadata to a common schema is similar to mapping and integrating schemas from different databases. As a result, we can borrow some instance-level based schema matching approaches, as is presented in [74], to help our mapping task using time series data. A detailed summary of these six time series based approaches can be seen in the Table 3.1 where features, models, metadata, evaluation strategies, and testbeds are listed.

To evaluate them consistently, we need to make sure they are compared to the same context. We can see the major differences among these approaches are: the features being extracted and the models being constructed, which are the subprocesses specified consisting metadata inference in Figure 3.1. Notice some other approaches utilize active learning to pre-cluster the data first to reduce the amount of required training data [40, 47]. We do not consider these steps in our evaluation as we treat the clustering based active learning approach as a technique to select and reduce training samples. The evaluation incorporating active approaches is left for future work. For all the models, we use seven widely used linear and nonlinear classifiers from column four in Table 3.1. In the end, we end up evaluating six types of features from six time series based metadata inference approaches using seven classifiers on each feature.

In addition to features and classifiers, we select same sets of BAS points and evaluation strategies to be applied to the same dataset. Specifically, we choose points driven by one application (APAR) to detect faults in AHU systems [15]. The

Year	Paper	Feature	Model	Metadata	Evaluation strategy	Testbed
1994	Li and Clifton [74]	Statistical quantities (mean, variance, coefficient of variation), then using SOM to reduce dimension and generate clusters	Trained back propagation network on one database with the cluster number as the label, then use the model to predict the cluster label for other databases	Similar attributes from heterogeneous databases	Test two databases with no common information and shows a low similarity; calculate the similarity between two groups of attributes and show similarities inside one database	3 pairs of existing databases
2015	Gao, Ploennigs and Bergés [42]	Statistical quantities (mean, median, quantiles, max, min.)	7 classifiers including kNN, Decision Trees, GNB, RBF SVM, Logistic Regression, AdaBoost, LDA	Types of points in BAS (VAV, AHU, FCU, electrical/light systems)	Stratified 20% train, 80% test, show TP/TN/FP/FN and F_1 score	2 buildings
2015	Hong et al.[41]	Statistical quantities (min, max, median, rms, quantiles, var, skew, kurt, slope) on 60-min sliding windowed (30-min overlapping), and then apply summary statistics (min, max, median, variance) on top of features	Combine the prediction from locally weighted classifiers, which are logistic regression, SVM and random forest	Types of points in BAS	Use one building to train and another to test	3 buildings
2015	Bhattacharya et al.[47]	Summary statistics (min, max, median, variance) on median and variance values of the 45-min sliding window (no overlapping)	Random forest	Properties of points in BAS (in addition to type)	Increase the examples to label and track the qualified sensors to be identified by the algorithm	3 buildings
2015	Balaji et al. [40]	1) min, max, mean, quartiles, range; 2) 3 Haar wavelets and 3 Fourier coefficients; 3) location and magnitude of top 2 components from piece-wise constant model, error variance; 4) first and second variance of difference between consecutive samples, max variance, number of up and down changes, edge entropy measure	Random forest	Types of points in BAS	Increase the number of labeled points of each type and check the accuracy	1 building[a]
2016	Koh et al.[49]	Mean, variance, dominant frequency, noise(error variance), skew, kurt	8 classifiers including GNB, LSVM, RF, RBF SVM, NN, DT, Adaboost, Bernoulli Naive Bayes (BNB)	Type, location and dependency	Use the co-location information of all VAV point types and the ground truth point types for one zone to infer point types from other zones	1 building

Table 3.1: A summary of six time series based inference approaches

[a] 4 buildings are included in the testbed while time series based approach is tested on 1 building.

effective implementation of ARAR has shown the ability to detect faults, reduce energy waste and bring many other benefits. As we analyze the BAS points in AHUs from different buildings, we are only concerned with one specific metadata property: the type of BAS points. It is worth pointing by type of BAS points, we refer to the concept-level properties of the metadata required by APAR.

3.3 Data

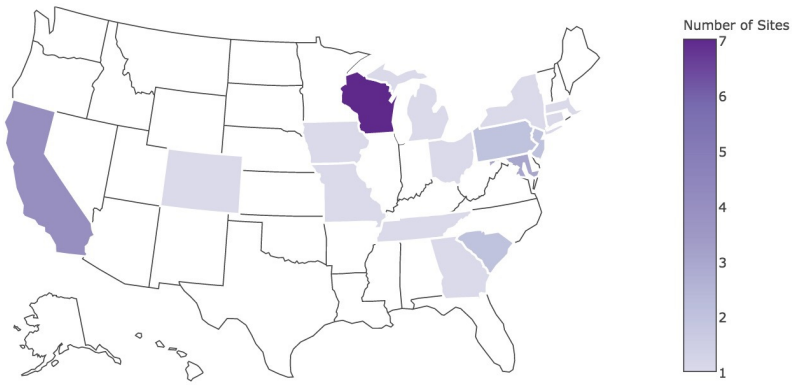


Figure 3.2: State-wise site distribution of AHU data in the United States

The data used for this study are collected using the data platform developed by Johnson Controls. We have access to AHU data from 421 building across 35 different sites. One site can be regarded as a group of buildings from one organization in a city. These sites encompass a wide variety of building types including educational institutes, office buildings, hospitals, libraries and others constructed in different years. Figure 3.2 shows the state-wise site distribution of the buildings we have data from, which covers different climate zones and 16 states. The data are collected for one year (from Jan. 1st, 2015 to Dec. 31st, 2015), including different types of points located inside AHUs. We ignore points which do not have one-year-long data. We choose one-year-long data to make sure the data collected show different thermal

conditions throughout the year.

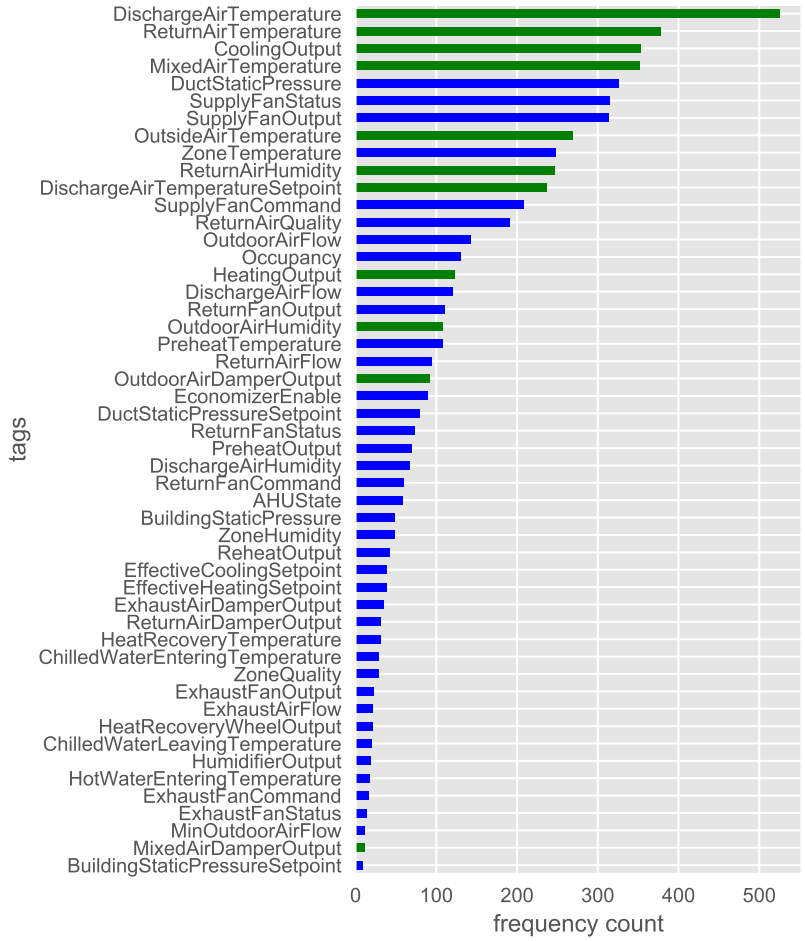


Figure 3.3: Frequency counts (greater than 30) of tags, green ones are selected by APAR

As different sensing points have distinct sampling intervals ranging from one second to one hour, we re-sampled all the points to 15 minutes intervals using padding by filling values forward. Additionally, we removed samples if they either had unclear descriptions or exhibited abnormal values (e.g., temperature values less than -50 Fahrenheit, or negative humidity values). More details about the data cleaning process can be found in Appendix B.1.1 This eventually gives us a raw data matrix X of size 6145×35040, representing 6145 BAS points with each having 35040 samples for the whole year (i.e., 1 sample every 15 minutes). For each BAS point,

a tag is attached to it following an internal convention inside the company that is considerably consistent. As shown by [38], the frequency with which the naming tags are used in buildings often follows an almost power law distribution. The top 50 frequent tags are shown in Figure 3.3. It is worth noting that these tags actually encode the metadata information including point types, physical quantities, medium, and functions. For example, "DischargeAirTemperatureSetpoint" represents a set point controlling the temperature of the air to be discharged out of an AHU.

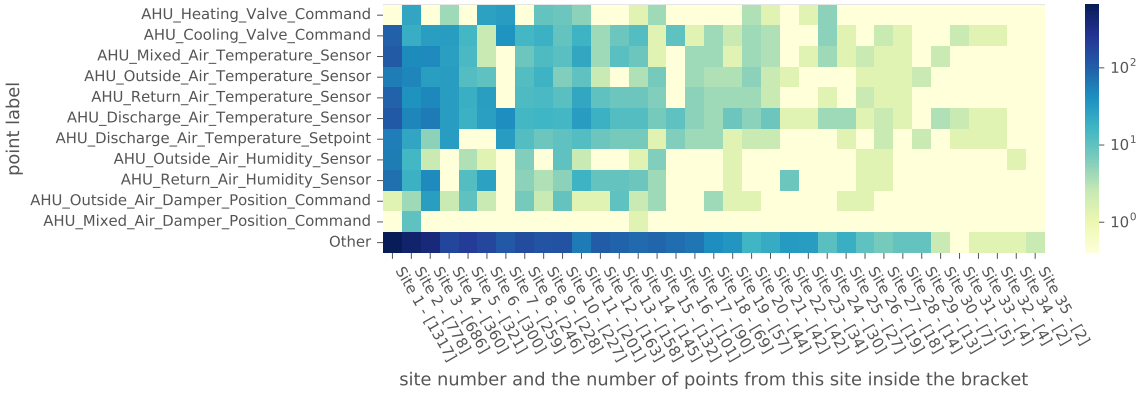


Figure 3.4: Frequency counts of each point label across 35 different sites, the number in the horizontal axis represents the total number of points at this site

Vendor tag names	Brick names
HeatingOutput	AHU_Heating_Valve_Command
CoolingOutput	AHU_Cooling_Valve_Command
MixedAirTemperature	AHU_Mixed_Air_Temperature_Sensor
OutsideAirTemperature	AHU_Outside_Air_Temperature_Sensor
ReturnAirTemperature	AHU_Return_Air_Temperature_Sensor
DischargeAirTemperature	AHU_Discharge_Air_Temperature_Sensor
DischargeAirTemperatureSetpoint	AHU_Discharge_Air_Temperature_Setpoint
OutdoorAirHumidity	AHU_Outside_Air_Humidity_Sensor
ReturnAirHumidity	AHU_Return_Air_Humidity_Sensor
OutdoorAirDamperOutput	AHU_Outside_Air_Damper_Position_Command
MixedAirDamperOutput	AHU_Mixed_Air_Damper_Position_Command
Other	Other

Table 3.2: Point name mappings between the vendor convention and Brick

As mentioned earlier, we focus on points required by APAR, which are marked

with green colors in Figure 3.3. It can be seen that about half of the frequent tags are selected by APAR. To map these points into a common schema, we choose to use Brick [29] here to map the required points. For the unselected points, we label them as "Other" as APAR does not require the metadata associated with those points. Another option could be to use all the labels during the training but focus on the points we are concerned with during the evaluation. However, the performance of models would drop by including more classes as it results in a more complicated decision boundary. As a result, we end up having 12 different types of point labels, as is seen in Table 3.2 where we have both the original vendor specific tag names and the Brick names. To better understand how these 12 different types of point labels spread over building sites, Figure 3.4 shows the number of counts of each label across sites, sorted from the site with most numbers to the least. We can see the distribution is quite unbalanced where some sites could have up to 1317 points, and some only have 2 points. Such a small number is likely due to the fact that old buildings still rely on the pneumatic systems and only a limited number of digital sensors are integrated into the BAS.

3.4 Experiments

In this section, we describe three sets of experiments using distinct evaluation strategies to answer the question about the generalizability of metadata inference approaches.

3.4.1 Generalizability on Single Site (S1)

To understand whether there is one inference approach that generally works well on each site, in this experiment, we train the model using a specific ratio of data on each site and test the model on the same site using the remaining data, and then

we iterate over all sites. The ratio of data to be trained is selected as 10% at first. We vary this ratio later to explore how the performance is affected. For training on each site using stratified 10% of data, we repeat the process 20 times to ensure coverage of the samples. We refer this experiment as Strategy 1 (S1).

This strategy envisions that, for any new unlabeled buildings, we can just label 10% of the BAS points and use this approach to infer the metadata for rest of the points. In this scenario, some sites may not have enough samples to use as 10% of training data, which means none of the classes have more than 10 points, and when this is the case, we ignore these sites. Additionally, for some sites, there are less than 10 points in certain classes, we ignore those classes and evaluate the approaches on data from the remaining classes.

3.4.2 Generalizability on Multiple Sites (S2)

To explore the generalizability on multiple sites, we conduct another experiment using leave-one-site-out cross-validation. That is, we use data from all but one sites to train and use the data from the remaining one site to test, and we iterate over sites. We refer to this experiment as Strategy 2 (S2).

This strategy makes sure that no data from the same site will appear in both training and testing samples. The reason for splitting by sites instead of buildings is to make sure we have enough test instances to evaluate. Such an evaluation can help us understand how the model performs on the unseen dataset. By using each of the sites as the testing site and observing the performance, we can reason whether the distribution drawn from a subset of buildings is generalizable. The vision is that we can use the trained model to predict the needed metadata for a new, unseen site.

3.4.3 Effects of Data (S3)

To study the effects of the amount of training data on the approaches, instead of using the whole-year-long data directly as we did in previous two strategies, we conduct a group of experiments varying the data being used to train the model. We refer to this as Strategy 3 (S3). Specifically, we consider the following four scenarios:

1) Varying the amount of data: we extend S1 by increasing the training ratio from 10% to 90% to study how the performance changes. Similarly, we extend S2 by changing the number of sites being used for training from 10 to 25 instead of 35;

2) Varying data duration: we use both weekly and monthly data to conduct the same analysis for S2 instead of using one-year-long samples;

3) Temporal effects: we further study how the model performs when training the model on one month and testing on another. We vary our data by site and month. For each month, we use any combination of 34 out of the 35 sites to train. We test on the remaining site for prediction performance over each of 12 months. We always train with one month of data and on 34 sites, and test on the remaining one site over all months;

4) Spatial effects: we also study the spatial effects of the data when we consider splitting data into different climate zones based on cooling degree days and heating degree days in the past 30 years as is seen in Table 3.3, which is defined by CBECS[2]. Since each zone contains different sets of points, to have a fair comparison across zones we synthetically generate a balanced data from the raw data where each zone has the same number of points for each point label.

[2]A spreadsheet file providing the climate zone for each US county can be found at `https://www.eia.gov/consumption/commercial/data/archive/cbecs/CBECS%20climate%20zones%20by%20county.xls`

[3]https://www.eia.gov/consumption/commercial/maps.php

Climate Zone	Cooling Degree Days	Heating Degree Days
cold	Fewer than 2,000	More than 7,000
cool	Fewer than 2,000	5,500 to 7,000
normal	Fewer than 2,000	4,000 to 5,499
warm	Fewer than 2,000	Fewer than 4,000
hot	2,000 or More	Fewer than 4,000

Table 3.3: Climate zone definitions according to CBECS[3]

Specifically, we first pick the point labels (classes) which have at least shown up 15 times within each climate zone (the number is selected so that we have a balance of the number of classes and the counts of samples). Once the labels are picked, we randomly draw 15 samples from each class without replacement for each zone. We end up having 105 points from 7 classes (15 per class) for each climate zone. Due to the limited number of points found in buildings that are in the hot zones, we only have data from four zones (cold, cool, normal, and warm).

To evaluate the spatial effects to the performance, for each zone we randomly use 50% of data from each class to train and test on the remaining 50% of data from this zone as well as all the data from rest zones.

3.5 Results and Discussions

In this section, we first define the metrics to be used, and then present results and discussions. The implementation details can be seen in B.1.

3.5.1 Metrics

Evaluating the performance of the multi-class classifier model is not a trivial task as there are many different metrics to choose with each depicting certain aspects of the model performance and there is no single best metric measure for model comparisons. Common choices of metrics include *single-class focus threshold metrics* such as

sensitivity/specificity, precision/recall, and F-measure, *multiple-class focus threshold metrics* such as accuracy, error rate, and kappa measures, and *ranking methods and metrics* such as receiver operation curve (ROC) analysis, precision-recall curves, and area under curve (AUC) [76]. Multiple-class focus metrics consider the overall performance and are less suited for the class-imbalanced situation as they are biased towards to the class with more samples [76]. Meanwhile, F-measure, a typical single-class focus metrics, is a popular metric in the information retrieval community and has been widely used for text classification due to the multiple classes and high class-imbalance nature of text datasets [77]. Typical ranking methods like ROC and AUC-based comparisons depict the trade-off between true positive rates and false alarm rates, and are less independent of the choice of classification threshold [78]. They also demonstrate advantages on datasets with skewed class distribution and unequal classification error costs [79].

When dealing with an unbalanced dataset, F_1 score and AUC are preferred. Nevertheless, both F_1 score and AUC are originally defined for binary classifiers. Extending these metrics for multiple classes requires averaging over the metric for each class[4]. Considering different averaging methods and assuming the distribution of each class (point type) in the real world is close to what we see in the data, we decide to use *micro* F_1 score to report the overall performance of the model, which is defined as follows.

[4]Averaging can be done using *macro*, *micro* or *weighted* strategies. The choice of the averaging depends on how each class is valued. The *macro* strategy calculates the unweighted mean, while *micro* uses the global quantities (e.g., precision, recall, true positives) to calculate the score and does not give advantages to small classes; and the *weighted* strategy calculates the weighted average, where weights correspond to the number of instances for each class. A *macro* average is more biased towards small classes and indicates the expected performance on a dataset with balanced classes. On the other hand, the *weighted* average is more biased to classes with more samples as it gives more weights to them. If we have a clear understanding of what weights we should assign to each class, we can also calculate a *weighted* average of binary metrics.

For predictions of class i out of C classes, for each fold/iteration j out of K folds/iterations, we calculate number of true positives ($TP_i^{(j)}$), number of false positives ($FP_i^{(j)}$), and number of false negatives ($FN_i^{(j)}$), by treating class i as positive and rest all negative. Then we calculate aggregated TP, FP, FN over each class i and each fold/iteration j, and define *micro* F_1 score as follows:

$$TP := \sum_{j=1}^{K} \sum_{i=1}^{C} TP_i^{(j)} \tag{3.1}$$

$$FP := \sum_{j=1}^{K} \sum_{i=1}^{C} FP_i^{(j)} \tag{3.2}$$

$$FN := \sum_{j=1}^{K} \sum_{i=1}^{C} FN_i^{(j)} \tag{3.3}$$

$$F_1 := \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.4}$$

This *micro* F_1 mathematically happens to be the same as the accuracy[5], which is defined as the total number of true positives divided by the total number of predictions. Since for each class i, counts of false positives from another class $\hat{i}$ will be counted towards false negatives of this class i and vice versa, all aggregated FP and FN in the definition above are counted twice when we are using them to define the total number of predictions:

$$TP + \frac{1}{2}FP + \frac{1}{2}FN \tag{3.5}$$

[5]An illustration example of different multi-class metrics showing *micro* F_1 and accuracy are equivalent can be seen at `https://github.com/INFERLab/metadata_inference/blob/master/multiclass_metric_test.ipynb`

As a result, the accuracy is defined as:

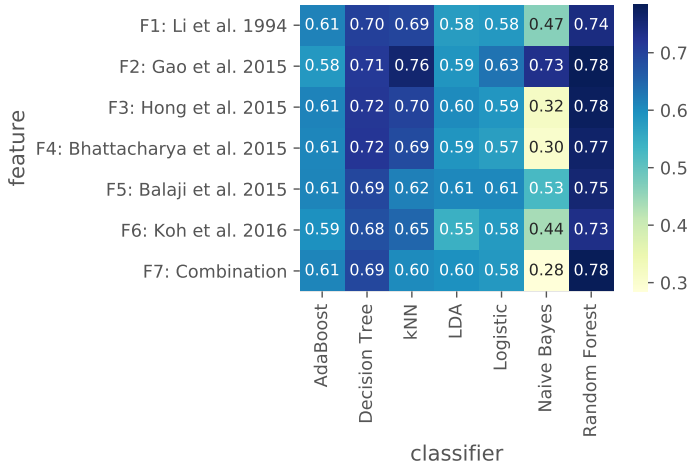
$$Acc := \frac{TP}{TP + \frac{1}{2}FP + \frac{1}{2}FN} == F_1 \tag{3.6}$$

Thus, we will use accuracy as the metric in this study. We do understand that using a single metric to describe a model is limited and could neglect other perspectives of the model, hence, we also provide the detailed performance of more other metrics including *macro* F_1 score and AUC score, as well as the single-class metrics including F_1 score, precision, recall, and AUC for each class in B.2, and we just use the accuracy simply as a way to compare the performance. In addition to all these different metrics, we also analyze the confusion matrix of the prediction to understand the reason for the misclassification.

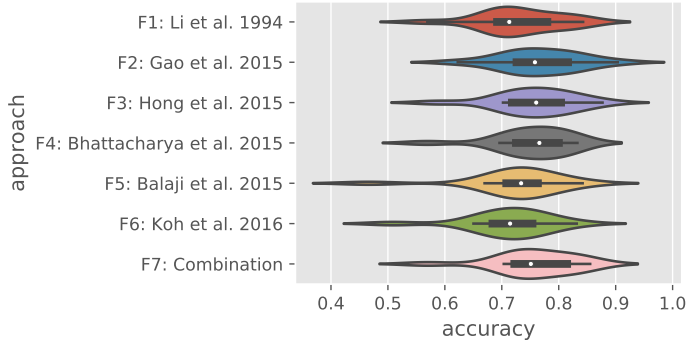
3.5.2 Generalizability on Single Site (S1)

Figure 3.5a shows the accuracy over 15 sites for each feature and classifier. As we can see, Random Forest outperforms the rest of the classifiers all the time, yielding the highest accuracy for each feature. It is worth pointing out that if we simply take the majority vote (i.e., always predict the most frequent class), the accuracy would be 56%. For datasets with a class imbalance, it is the improvement over this majority vote what matters. So a 78% accuracy, as obtained with the combined features in our study, indicates a significant improvement over the base case of using the majority vote.

To understand how each feature performs over sites, Figure 3.5b shows the violin plot of accuracy score over 15 sites for different features using Random Forest. The score does vary drastically across sites for the same feature, with the difference between the maximum and the minimum being 40% to 60%.



(a) Accuracy score matrix



(b) Violin plot over 15 sites for different features using Random Forest

Figure 3.5: Violin plot of accuracy score and accuracy score matrix for different features and classifiers (S1)

The result in Figure 3.5b indicates that the same metadata inference approach can perform quite differently on different sites with a standard deviation from 0.07 to 0.09. This variance is due to the distinct behaviors of points on each site. Additionally, all features show close performance as they are all similar in the sense that they are based on descriptive statistics (e.g., max, mean, median, etc. of the time series). We conduct the Kruskal-Wallis H test [80] to test whether accuracy scores over sites from each approach are drawn from the same distribution. The resulting p-value is $p \ll 0.001$, indicating that there is not enough evidence to reject the null

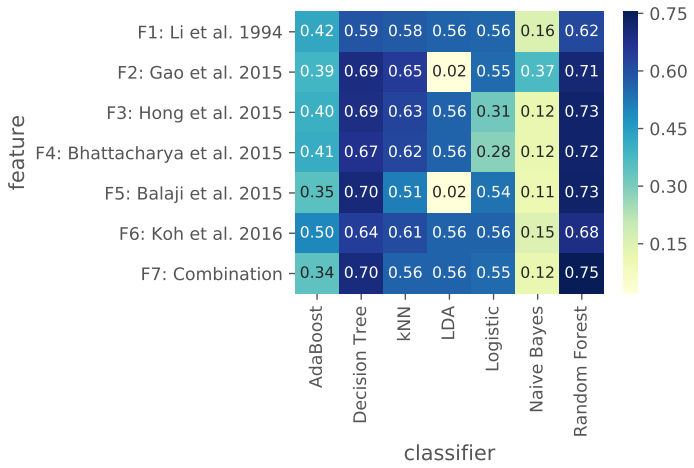
hypothesis that scores generated from different approaches are from the same distribution. When we examine the feature for each site yielding the highest accuracy, we find that almost all features achieve their highest site-specific performance using Random Forest. Moreover, for any fixed feature, Random Forest outperforms the rest of the classifiers all the time, as shown in the last column of Figure 3.5a. This signals that Random Forest is well suited for classifying point types in buildings due to its capabilities in dealing with flexible and overlapping decision boundaries and noisy data, which is also aligned with our prior research results [42].

This experimental result implies that it is feasible to select a building site, label 10% of metadata for each point type, train a model using inference approaches, and we are expected to label 78% of the rest points with consistent metadata correctly. However, the actual performance can vary depending on which specific building site is being used.

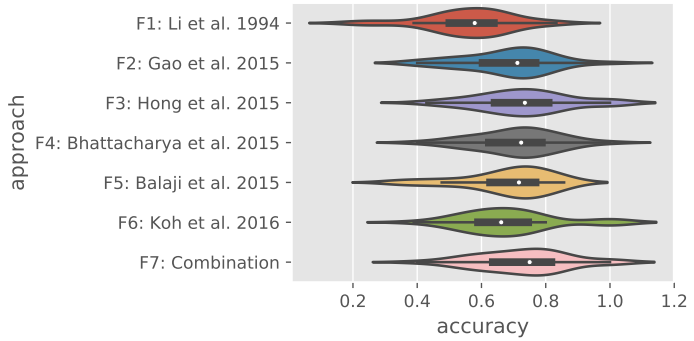
3.5.3 Generalizability on Multiple Sites (S2)

To summarize the experimental results of S2, where the goal is to evaluate the inference performance of the model on unseen buildings based on training data from well-labeled buildings, we compute the accuracy matrix of different features across different classifiers. The results are shown in Figure 3.6a. We also show the violin plots of the accuracy scores over 35 iterations of test sites in Figure 3.6b. As is expected, all statistical-based features achieve similar results with Random Forest being the best classifier.

On average, the scores from S2 are slightly lower than those from S1. Part of the reason is that S2 is using a stricter condition where the test building sites do not overlap with the training sites. The standard deviation of the accuracy score across sites is also more significant for S2 ($\sim$0.18) as compared with S1 ($\sim$0.09).



(a) Accuracy score matrix



(b) Violin plot over 35 sites for different features using Random Forest

Figure 3.6: Violin plot of accuracy score and accuracy score matrix for different features and classifiers (S2)

This makes sense, given that the variation in S2 is stronger due to the disjoint training and testing samples, as well as the increased number of sites. Similarly, we conduct the Kruskal-Wallis H test on the 35 accuracy scores from each approach and obtain a p-value of $p < 0.001$, again failing to reject the null hypothesis that the distributions are the same. We also notice the performance difference between these two strategies is not remarkable, which might imply that the information from a subset of buildings is capable of representing the distribution of the statistical features being derived from each point type using the historical time series of another

group of buildings. This indicates that time series values associated with points from multiple buildings could have similar distributions, which is of particular interest as it shows the possibility of training a model on some buildings and using the model for other unseen buildings. However, it is worth noting that this initial finding is based on points in AHUs from buildings within one vendor's portfolio. The validity of the conclusion remains to be evaluated on more diverse building portfolios.

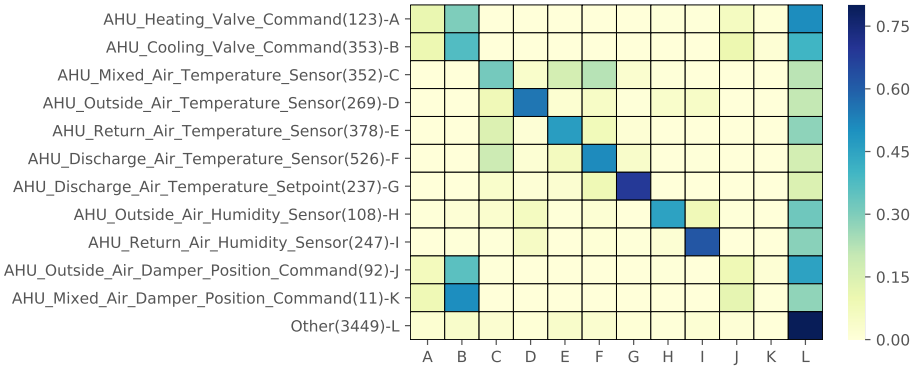


Figure 3.7: Normalized confusion matrix by row using F7: Combination and Random Forest (S2). The number inside the bracket beside the label name on the vertical axis represents the number of testing instances for this class

To further understand how the approaches perform under S2, we look at the confusion matrix using "F7: Combination" and Random Forest to see which predictions are incorrect. Due to the unbalanced number of samples for each class, we show a normalized confusion matrix (i.e., each element is divided by the sum of all the elements in the corresponding row). The values in each row represent the average probability vector of this type being predicted for each of 12 types in Figure 3.7. The number inside the parentheses beside the label name on the vertical axis represents the number of testing instances for this class. We can see that "Outside Air Damper Position Command" and "Mixed Air Damper Position Command" are most easily confused with "Cooling Valve Command". This is a reasonable mistake, as they are all generating values within the range [0,100] and the damper output values are

strongly correlated with the cooling status, which can impact each other and show similar behaviors. The same result can be seen in Table B.3 where we compute the precision, recall and F_1 score for each class. We also notice that many point labels are misclassified as "Other", which is due to the diverse behavior of excluded points in AHUs. If we can somehow exclude all "Other" points from the analysis and only focus on the selected 11 types of point labels, the accuracy score increases to 80% using S2.

3.5.4 Effects of Data (S3)

For this subsection, we explore how the model performs under S3 where we consider varying the amount of data used for training the model as well as the duration represented in the data (e.g., a full year of measurements), the seasons that are represented, as well as other temporal and spatial effects.

Amount of Data

We first explore how the accuracy changes when we vary the amount of training data for S1 and S2 while keeping the temporal duration (1 year) of each sample fixed and not paying attention to the spatial location of the buildings in the training sample. For S1, we increase the training ratio from 10% to 90% using "F4: Hong et al. 2015" and Random Forest [6]. As is expected, the accuracy increases from 78% to 90%.

Similarly, for S2, we vary the number of sites being used. We start with only ten sites, and we use the "leave-one-site-out" strategy to evaluate the performance. By adding more sites to the model, we want to find out how the performance changes.

[6] If not specified, the following explorations of data effects are all using this feature and classifier as shown in earlier results. The choice of feature does not significantly affect the result as long as it is one of the statistically based features, which summarize the descriptive statistics of the time series.

Each time, a number of sites are randomly chosen out of 35 sites, and the process is iterated 20 times. We pick the number of sites to vary from 10 to 25 since the number of possible combinations for choosing 10 out of 35 is the same as choosing 25 out of 35. We then calculate accuracy score over 20 iterations as the performance metric. Figure 3.8 shows how it changes when we vary the number of sites. As we see, the general trend of the accuracy score is slightly increasing, and the standard deviation decreases, indicating that the model becomes more accurate and stable when we have data from more building sites.

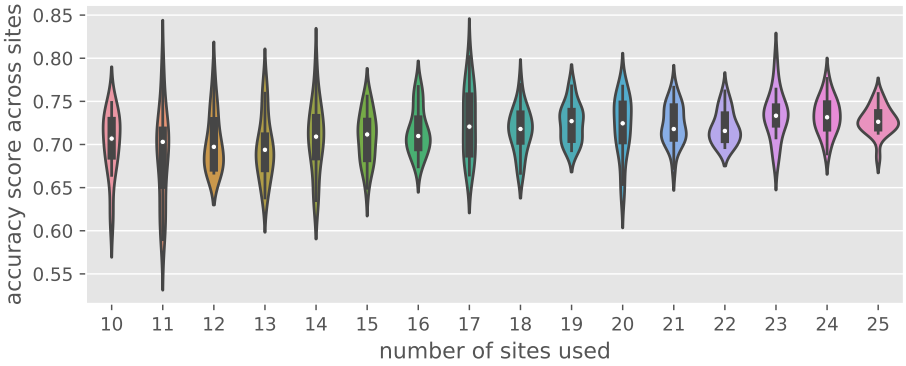


Figure 3.8: Violin plot of accuracy change when we vary the number of sites (S2)

Duration of Data

To study the effects of the duration of the data used for training, we divide the year-long data into week-long and month-long segments and evaluate the model performance for each segment using S2 where we train the model using data from 34 sites and test on the data from the remaining site, and we iterate until each site has been used as the test site once. The evaluation gives us 52 values of the accuracy score on each testing site for weekly data and 12 values for monthly data. Table 3.4 summarizes the result of the accuracy score from data of different durations. For the yearly case, we report the statistics for accuracy score given 35 iterations on each

test site. As is seen, the yearly data provides slightly better performance compared with others, which makes sense since a longer duration can capture more temporal characteristics of point behaviors. Given the performance drop for the weekly data is not significant, we may still be able to use metadata inference approaches with a short duration of data when one-year-long data are not available.

accuracy	year	month	week
mean	0.735	0.701	0.671
median	0.739	0.687	0.662
standard deviation	0.158	0.155	0.158

Table 3.4: Statistics of accuracy when using data from different durations

Temporal Effects

To study other temporal effects, we report the average accuracy score across all testing sites for all possible pairs of training and testing months.

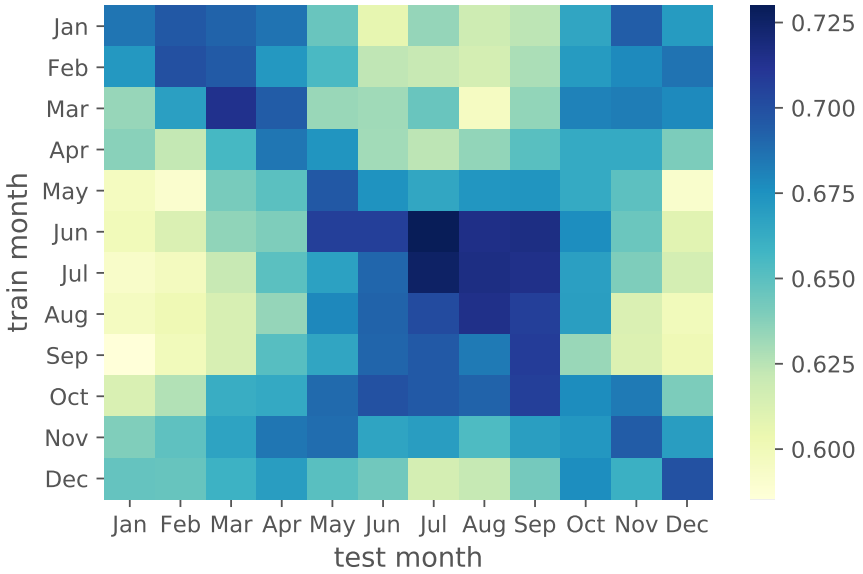


Figure 3.9: Accuracy score when training on one month and testing on another

Figure 3.9 shows the average performance of training on one month and testing on another month. If we sum the values in the diagonal and take the mean, the

value should be close to the mean of the monthly result 0.701 in Table 3.4. The results from Figure 3.9 indicate that training and testing on the adjacent months are likely to produce a slightly better performance. This implies that when training and testing models on different building sites, it is not necessary to make sure the data are from the same temporal period. The model will generally perform well as long as the data are temporally adjacent.

Spatial Effects

We also wanted to explore how the model performs when we consider spatial differences and split the data into different climate zones. Specifically, we iterate the experiment process (as is described in Section 3.4.3) 20 times for each zone on the synthetic dataset and report the average accuracy when training on one climate zone and testing on another.

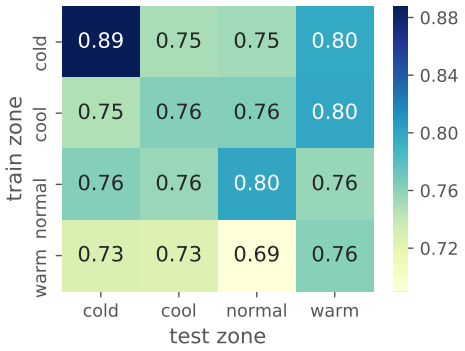


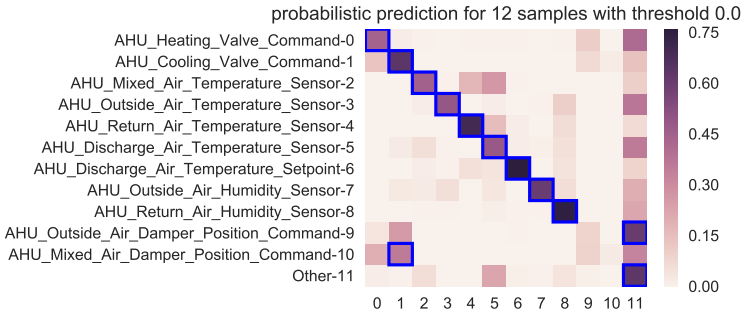
Figure 3.10: Accuracy score when training from one climate zone and testing on another

Figure 3.10 shows the performance of training on one climate zone and testing on another zone. We can see the performance is slightly better within each zone compared with training and testing across zones. Training on data from cold zones tends to provide better results. Furthermore, if we check the variations of each experiment, the standard deviation is between 0.02 and 0.06, which means the difference between training and testing on different zones is not that significant. This is

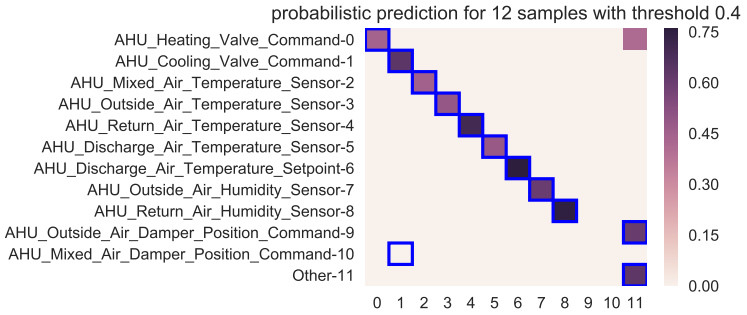
also aligned with the conclusion we drew previously in S2 that the time series values associated with points from different buildings have similar distributions, regardless of the location of the building.

3.5.5 Probability Perspective

So far, we have been interpreting prediction results deterministically. However, another interesting perspective is to look at the predicted probability mass vector. In other words, for each time series, the predicted output is not a simple label; instead, we have a vector indicating the probability that this time series belongs to each class.



(a) Raw prediction, the blue square box highlights the most likely prediction



(b) Setting .4 cutting threshold, the blue square box highlights the most likely prediction

Figure 3.11: An example illustrating the probability prediction metric

Figure 3.11 explains the idea using 12 instances (one from each class). Each row in Figure 3.11a is a probability mass vector indicating the likelihood of this point

belonging to each class. The ideal prediction occurs when the most likely predictions for each vector fall on the diagonal. This is similar to Figure 3.7 where we show the average probability for each class, while Figure 3.11a represents the probability for each specific point.

Given a probability threshold p ($0 < p < 1$), we have N time series $\mathbf{X} \in \mathbb{R}^{N \times T}$ being predicted as $\mathbf{Y} \in \mathbb{R}^{N \times m}$ where m is the number of possible classes. For each prediction $y_i \in \mathbb{R}^m$, we count it as part of the covered prediction set $\mathcal{Y}_p^{cover}$ if $\max(y_i) > p$, and count it as part of the uncertain prediction set $\mathcal{Y}_p^{uncertain}$ if $\max(y_i) \leq p$. Figure 3.11b shows the case when we set 0.4 as a threshold. In other words, the covered prediction set includes predictions with more confidence, and the uncertain prediction set includes predictions with more uncertainties.

Then, we can define the following two metrics given probability threshold p:

coverage: percentage of predictions with confidence higher than p

$$\frac{\left|\mathcal{Y}_p^{cover}\right|}{N} \tag{3.7}$$

coverage accuracy: percentage of correct predictions among covered set

$$\frac{\sum_{y_i \in \mathcal{Y}_p^{cover}} \mathbb{1}(y_i^{true} = \arg\max_{j=1,\cdots,m} y_{ij})}{\left|\mathcal{Y}_p^{cover}\right|} \tag{3.8}$$

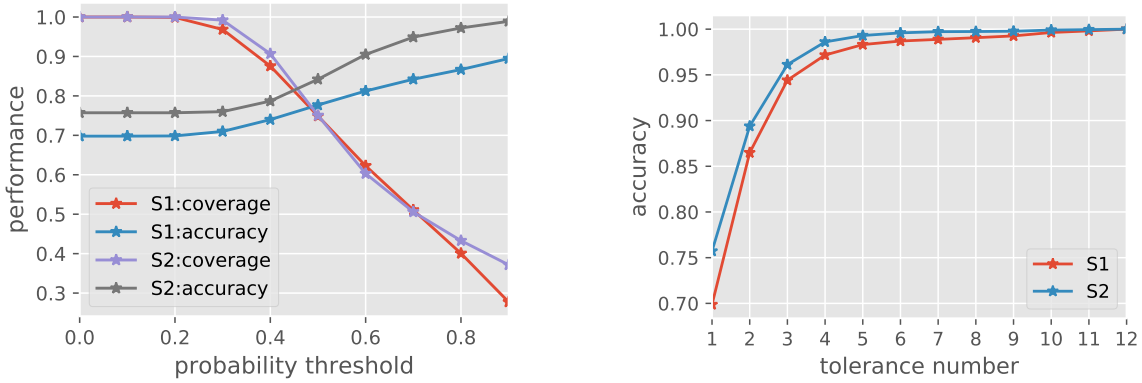
Additionally, if we tolerate mistakes generated by the probability prediction and assume that the predictions are correct as long as the actual label is within the top d predictions ranked by probability vector, we can define another metric given the tolerance number d:

tolerance accuracy: denote $\hat{y}_i^d$ as the top d predictions ranked by probability

vector y_i, the accuracy when we tolerate d mistakes is

$$\frac{\sum_{i=i}^{N} \mathbb{1}(y_i^{true} \in \hat{y}_i^d)}{N} \tag{3.9}$$

In the example shown in Figure 3.11a, the original accuracy is 83%(10/12). However, we can have an accuracy of 90.9% (10/11) with 92% (11/12) coverage by setting up 40% as the probability threshold; and the tolerance accuracy being 92%(11/12) by setting the tolerance number to 3 ($\hat{y}_9^3$ contains the true label and $\hat{y}_{10}^3$ does not).



(a) Coverage and accuracy

(b) Tolerance accuracy

Figure 3.12: Probability metrics from two perspectives

Using the definitions above, we calculate these metrics by varying the tolerance number and the probability threshold for both S1 and S2 in Figure 3.12. As we can see the tolerance accuracy can go up to 95% if we tolerate three guesses. The use case for this is to reduce the labeling efforts from identifying 1 out of 12 different labels to identifying only 1 out of 3. Another perspective is to set up the probability threshold. By setting it to .6 for example, we can cover 60% of the points with an accuracy up to 80% - 90%. If we want to be more aggressive, we can choose to only cover 40% of the points with an accuracy up to 95% in the case for S2.

This indicates we can trust the algorithm with high probability (up to 95%) to label 40% of the total data correctly, and that we would only need to manually label the remaining 60%. By incorporating probabilistic perspectives into the predictions, it can be more efficient for building operators and managers to produce the consistent metadata for buildings in practice.

3.6 Conclusion

This chapter investigates the generalizability of six time series based metadata inference approaches by evaluating them on sensors from 614 AHUs and studying how the data used to train the models can affect their performance. We find that when evaluating the approaches on such a dataset, we can achieve the best performance with an accuracy of 75%, regardless of training and testing on the same site (S1: 10% to train, 90% to test) or training and testing on different sites (S2: leave-one-site-out cross-validation). Moreover, these different testing approaches do not exhibit a significant difference in terms of performance.

Another way to interpret these results is as follows. Consider ten building sites containing a total of 1000 distinct BAS points, where each site has 100 points. If we can obtain trust-worthy labels for at least 10 of these points in each site and use them to train the model, existing metadata inference approaches could impute the rest of the labels (i.e., the remaining 90 points on each site) with 78% accuracy on that same site. When we are training and testing on different sites, the result indicates that we can randomly pick nine sites to use 900 points to train the model, and we are expected to predict 75% (75 points) of 100 points from an unseen site correctly. At first glance, it may seem as if training and testing on different sites require more training data, but it does not require any training data for a new

unseen site, which would reduce the amount of training effort significantly as the number of testing sites increases.

To study the feasibility of these approaches in more realistic conditions, we explore proxies for the amount of human effort required to train the models, including varying the amount, duration and temporal/spatial factors of the training data. We find that by increasing the training ratio from 10% to 90%, we can improve the accuracy score from 78% to 90% when training and testing on the same site.

Increasing the amount of data being used also helps to reduce the variance of the performance of the model in the case of training and testing on different sites. We also observe that yearly data show the strongest patterns to differentiate distinct point labels. By using training and testing data from different time periods, we find the model can generally perform well as long as the data are temporally adjacent. However, when we pick data from different climate zones, we have not found training and testing on similar climate zones can provide significantly better results other than when using data from cold zones, indicating the spatial effects to the model are smaller compared with the temporal effects.

Additionally, we define metrics including coverage, coverage accuracy, and tolerance accuracy from probability perspectives. These metrics can make the predictions of the metadata from the model more useful for building operators and managers, as they can reduce the amount of time to focus on the points selected by the model. For instance, direct predictions can only label 75% of points correctly. However, with probability perspectives, we can predict 40% of points with a very high accuracy up to 95%, and for the remaining 60% of points, we can reduce the searching efforts from 12 different types to 3 most likely candidates.

Several future working directions are suggested in this research field. First, more

advanced feature extraction techniques considering temporal evolution and multivariate relationships of BAS points should be studied to differentiate inseparable points by simple statistical features. These could include autoregressive-moving-average models, graph and network analysis of sensor nodes, etc. Secondly, a more comprehensive representation of metadata needs to be reasoned from existing BAS on a large scale in addition to the types of BAS points, such as the location of the points, the equipment the point belongs to, the functions and interactions between sensors and building components. All these research directions will lead us towards an automated metadata standardization in BAS to facilitate the ultimate vision of portable FDD applications.

Chapter 4

Convolutional Neural Networks Applied to Metadata Inference

In this chapter, we explore a new metadata inference approach to infer the type of BAS points from time series data based on convolutional neural networks. The purpose is to investigate the inference problem from a purely data-driven perspective where the efforts to design hand-crafted features are avoided. We want to explore whether the neural network based approaches can achieve the same or even better performance compared with existing approaches. Additionally, we also explore the feasibility of using hierarchical classifiers and ensemble classifiers to further improve the performance of existing models.

4.1 Motivation and Related Work

In the specific domain of time series based sensor metadata inference for BAS, many approaches have been proposed using different features [44, 42, 45, 46]. These approaches use hand-crafted engineered features (e.g., descriptive statistics, maximum, minimum and standard deviation) and very often overlook the sequential

information that can be extracted. To illustrate the potential problem caused by this, Figure 4.1 shows the plots of three datasets from Anscombe's quartet [81]. If we consider the horizontal axis as the time, we can see the three datasets have different patterns. However, when we calculate some statistical properties, all three datasets have almost the same values as seen in Table 4.1.

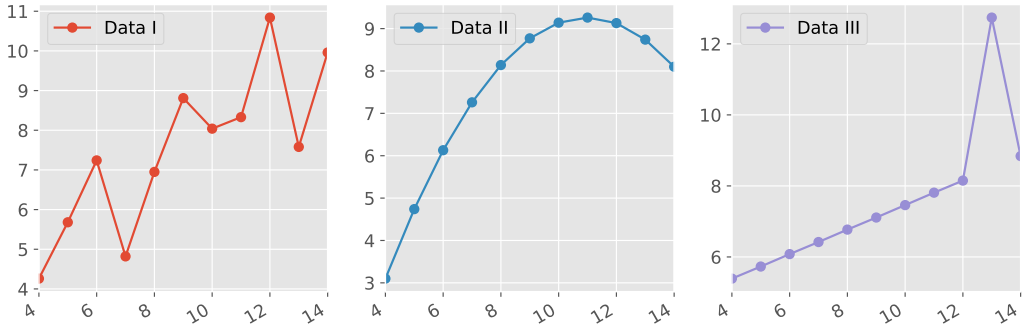


Figure 4.1: Plots of three datasets from Anscombe's quartet

Statistical Property	**Value**	**Accuracy**
Mean	7.50	to 2 decimal
Sample variance	4.125	+/- 0.003
Correlation between x and y	0.816	to 3 decimal
Linear regression line	$y = 3.00 + 0.500x$	to 2 and 3 decimal, respectively
Coefficient of determination of the linear regression	0.67	to 2 decimal

Table 4.1: Statistical quantities for three datasets

This phenomenon is further explained in [82]. In short, different datasets with varying appearance could have the identical statistics [1], which makes such statistical features less sensitive to the change of the sequence (order) of values. Although researchers have tried to overcome such issues by dividing time series data into multiple windows, extracting features from each window, and then taking another

[1] An animation can be seen at `https://www.autodeskresearch.com/publications/samestats`

summary statistics of features from multiple windows [41], the power of these features to incorporate the sequential information could still be limited.

To incorporate the sequential information, other efforts have been made to exploit deep neural networks, especially convolutional neural networks (CNN) for end-to-end time series classification [83]. With different processing units (e.g., convolution, pooling, rectifiers), CNNs have shown success in computer vision, natural language processing, speech recognition, and time series analysis [84]. CNNs have been mostly used as a supervised classification model when initially being designed. However, one special architecture of CNN has been proposed as an unsupervised feature extraction method directly using an auto-encoder (AE) structure [85]. Convolutional neural network auto-encoders (CAE) learn how to map the original data into a latent representation (encoding process) which is then mapped back to the original data (decoding process) using a convolutional layer in the middle. This latent representation is normally used as the feature of the original time series. Due to the convolution operations performed by continuously sliding windows of different scales to the time series, the sequential information is preserved in the latent layer. CAE has several variations including pooling and unpooling operations [86], convolution and deconvolution operatoins [87], tied weights for encoder and decoder layers [88], predicting noise as targets instead of the original inputs [89], etc. As CAE reduces the efforts to build hand-crafted engineered features and can incorporate the sequential information, we attempt to build a specific architecture of CAE for the purpose of inferring sensor metadata from time series in buildings. Additionally, as a comparison of supervised method versus unsupervised feature extraction methods, we will also build a CNN as a classier directly. The detailed description of the proposed method will be presented in the next section.

4.2 Methodology

To describe the methodology in detail, we will start by defining the problem of time series classification using more specific notation. Given N one-dimensional time series of length T from N sensors $X^{N\times T} = \{x_1, \cdots, x_i, \cdots, x_N\}$, where $x_i \in \mathbb{R}^T$ and the corresponding class labels are $Y^{N\times 1} = \{y_1, \cdots, y_i, \cdots, y_N\}$ and $y_i \in \{1, 2, \cdots, C\}$ (C is the number of unique classes), the objective is to predict the class labels Y based on time series data X.

Suppose we have a function, or a model f, which is able to map x to y. Denote $\hat{y}_i = f(x_i)$ representing the mapping relationship. The performance of the model can be quantified by comparing the predicted label $\hat{y}_i$ with the true label y_i using a loss function h. One example loss function can be defined in terms of the zero-one loss using the indicator function:

$$h(y, \hat{y}) = \mathbb{1}(y = \hat{y}) \tag{4.1}$$

If we evaluate different models from a model set $\mathcal{F}$ using N time series, the optimal model can be found through the following optimization problem:

$$f^* = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{N} h(y_i, f(x_i)) \tag{4.2}$$

This model f is typically trained using a portion of labeled data (both x and y are given) and then evaluated on the remaining unlabeled data (only x is given). The model involves two parts, namely feature extraction and classification. Feature extraction aims to find the feature, which is another representation of the original data X, that allows the classifier to better discriminate data of different types. Depending on the underlying assumptions of the data, various strategies can be

used to build the model with distinct features and diverse classifiers.

In the previous chapters, we have extracted different statistical based features and use the random forest, which is the classifier yielding the best performance. In this chapter, we will explore other possibilities of building this model using convolutional neural networks based approaches. We describe two such models, one is a model named **cnn-clf** that can conduct both feature extraction and classification, and the other is a feature extraction method named **caeF**. Both of these methods are explained in more detail in the next section.

4.2.1 Convolution Neural Network as a Classifier

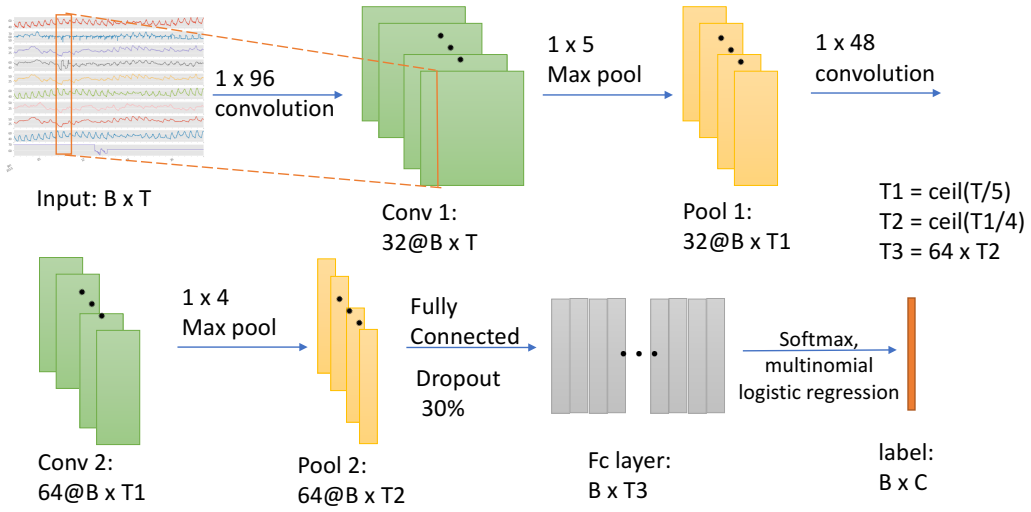


Figure 4.2: Architecture of the convolutional neural network for time series data classification

In this approach, the CNN is used as a supervised classifier on raw time series data directly. The architecture can be seen in Figure 4.2 where we feed data with batch size B of dimension T. We build the network with two convolutional layers and two pooling layers followed by a fully connected layer with 30% drop-out ratio. The number of convolutional filters and the size for convolution as well as the pooling

can be seen in the figure as well. The last layer is based a softmax function to map the continuous variables to C discrete labels. This model can be trained from the data with known labels and then used to infer the metadata for points with unknown labels.

In the implementation phase, we use a batch size of 200 with the dimension of the data being 2976 (one-month-long) to predict 20 different classes. The training and testing strategy will be discussed in Section 4.4. We mark this approach as **cnn-clf**.

4.2.2 Convolution Neural Network Auto Encoder

A simple one hidden layer auto-encoder takes an input $x_i \in \mathbb{R}^T$ and maps it to a latent representation $h_i \in \mathbb{R}^d$ using an encoder function $h_i = f_E(x_i) = \sigma(Wx_i + b)$ where $W \in \mathbb{R}^{d \times T}$ and $b \in \mathbb{R}^d$ are the weight and bias parameters respectively, and $\sigma(\cdot)$ is an activation function to regulate the extent the neural in the specific layers to be activated[2]. The latent representation h_i is then mapped back to the reconstructed input $\hat{x}_i \in \mathbb{R}^T$ using an decoder function $\hat{x}_i = g_D(h_i) = \sigma(W^{'}h_i + b^{'})$ where weights (W) are normally tied with the parameters from the symmetric encoder layer forcing $W^T = W^{'}$. This reduces the number of parameters to train and regularizes the model to be simple. By minimizing the following loss among all samples iteratively, we can find the optimal weight and bias parameters that minimize the reconstruction error:

$$Loss = \sum_{i=1}^{N} ||x_i - \hat{x}_i||_2 = \sum_{i=1}^{N} || \{x_i - g_D [f_E(x_i)]\} ||_2 \tag{4.3}$$

Normally, the auto-encoder can have multiple encoder and decoder layers which allows one to learn a deeper representation. The loss function can be represented

[2]The activation function σ is normally in the form of sigmoid, tanh or ReLU.

as follows with a chain structure if we have m encoder layers and decode layers:

$$Loss = \sum_{i=1}^{N} || \left[x_i - g_D^1(g_D^2(\cdots g_D^m(f_E^m(\cdots(f_E^1(x_i)))))) \right] ||_2 := ||X - \mathcal{D}_\phi(\mathcal{E}_\theta(X))||_F \tag{4.4}$$

where $\mathcal{D}$ and $\mathcal{E}$ are notations used to represent all decoder and encoder layers respectively, with weights and biases denoted as ϕ and θ.

CAE essentially has the same structure as a regular auto-encoder with the difference being that the encoder function is based on convolution and pooling operations and the decoder function is based on deconvolution and unpooling operations. A good explanation with 2D images can be found in [87]. An example of the CNN architecture for time series can be seen in Figure 4.3.

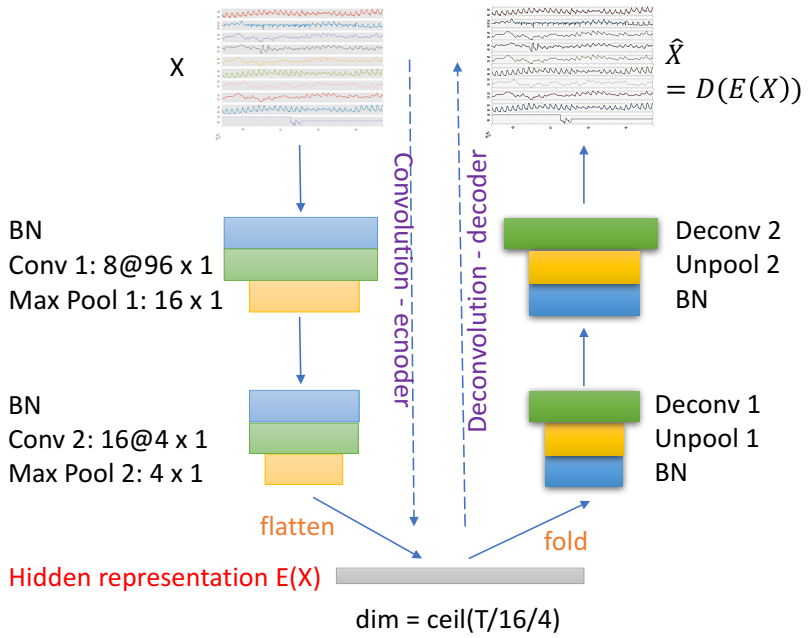


Figure 4.3: Architecture of the convolutional neural network autoencoder for feature extraction

To explain the architecture, we use an auto-encoder with two encoder layers

and two decoder layers. The structure of the layers has the transformation from $h_i^{E1} = f_{E1}(x_i), h_i = f_{E2}(h_i^{E1}), h_i^{D2} = g_{D2}(h_i), \hat{x}_i = g_{D1}(h_i^{D2})$. Suppose we are using k filters to apply the convolution on input X with a stride of 1 and padding 0 to make sure each filter slides T times, then we will have h^{E1} of dimension $N \times T \times k$, and $h_{i,k}^{E1} = \sigma(x_i \star W^k + b^k)$ where $\star$ is the convolution operation by sliding the window on the data and take the weighted summation. A good illustration of common operations including convolution, deconvolution, pooling and unpooling can be seen in [87].

In addition to these typical operations for CAE, we also adopt the batch normalization (BN) technique to avoid the problem of vanishing gradients as is suggested in [90]. Due to the fast speed of rectified linear units (ReLU), we will use it as the σ activation function after the convolution operation. The weights will also be tied to the encoder and decode layers. However, since there exist negative values in the time series data and ReLU will force them to be zero, we will not apply any activation for the last layer in order to reconstruct the original input. Hence, the weights on the last layer will not be tied with weights from the first layer while the weights on the rest layer are tied in a symmetric fashion.

Once the network is trained, we can use the latent representation $f_E^m(\cdot)$ in the hidden layer as the feature on which to perform classification. In the implementation phase, we use the same parameter for the convolutional layers and pooling layers specified in the figure. We mark this feature **caeF**. Such feature incorporating sequential information will be evaluated on a classifier to compare with the existing hand-crafted engineered features.

4.2.3 Baseline Approach

Instead of using features based on existing approaches directly as the baseline, we summarize the features into several categories based on the literature. The details of the implementation can be seen in Appendix C.1. The categories are:

- Statistical feature (**statF**): Descriptive statistics such as mean, median, standard deviation, etc. of the time series.
- Window feature (**winF**): We divide the data into multiple sliding windows and calculate features within each window. For each feature calculated over multiple windows, another statistics can be used to generate a higher level of abstraction.
- Time-frequency feature (**tfaF**): Features derived from time-frequency analysis information including fast Fourier transform (FFT) and wavelet analysis.
- Distance-based similarity feature (**dtwF**): We use dynamic time warping (DTW) as a distance measure to quantify the similarity between any pair of time series.

Additionally, we will also concatenate the above features to produce a combined feature (**combF**). For each of the features above, the random forest will be used as the classification model, and the same parameters will be used, as is shown in Table B.1.

It is worth noting that the above categories of features could have overlaps, for example, STFT in time-frequency analysis can also be considered as a window feature. In this chapter, by saying **winF**, we mean applying statistics on windows, and by saying time-frequency feature, we mean applying Fourier transform and wavelet decomposition on the whole time series without using windows. Also, for

the combined feature, we will simply combine all of the features. The study of mixing and combining different features is not the focus of this work.

To summarize, we will use five approaches based on existing literatures including **statF**, **winF**, **tfaF**, **dtwF** and **combF**, and two approaches based on convolutional neural networks namely **cnn-clf** and **caeF**. These approaches are all feature extractions methods which will be combined with random forest to make predictions except for **cnn-clf** which can classify point types directly.

4.3 Data

We use the same dataset that is described in Section 3.3. However, we label the points differently. In the previous chapter, we focus on inferring 11 point types that are required by a particular FDD approach - APAR. In this chapter, we investigate the performance of metadata inference on the commonly required BAS points by FDD approaches, which also align with 20 most common point types that appear in AHUs. These top 20 frequent BAS points are shown in Figure 4.4. Meanwhile, we eliminate the points whose frequency counts are less than 100. Moreover, we only use one-month-long data from January in this study. This eventually gives us a raw data matrix X of size 4822×2976, representing 4822 BAS points with each having 2976 samples in January of the year 2015 (i.e., 1 sample every 15 minutes).

To better understand how these 20 different types of point labels spread over building sites, Figure 4.5 shows the number of counts of each label across sites, sorted from the site with most numbers to the least. We can see the distribution is quite unbalanced with some sites having up to 1247 points and some only have 1 point.

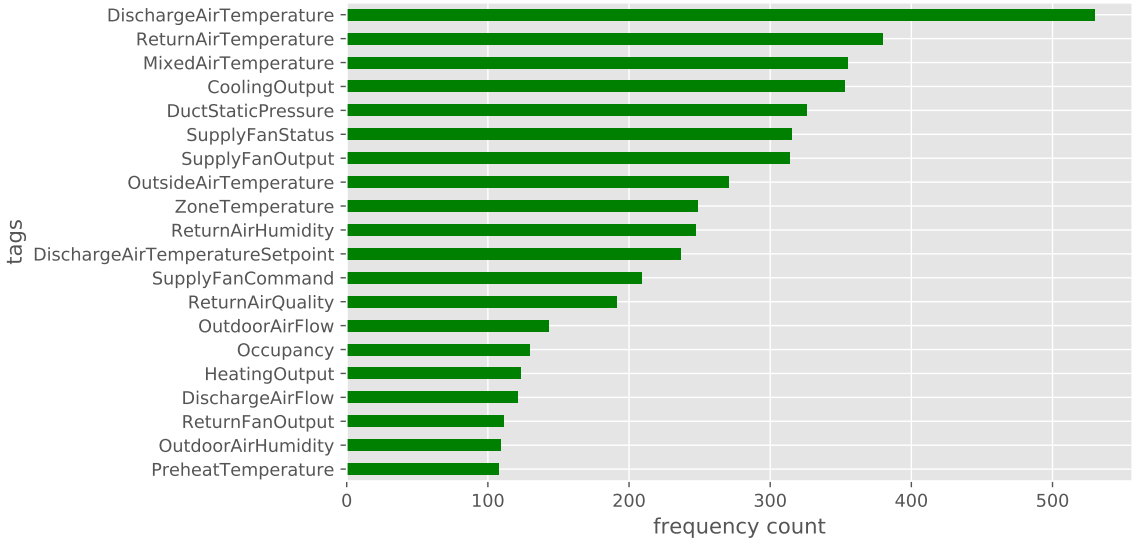


Figure 4.4: Frequency counts (greater than 30) of tags, green ones are selected by APAR

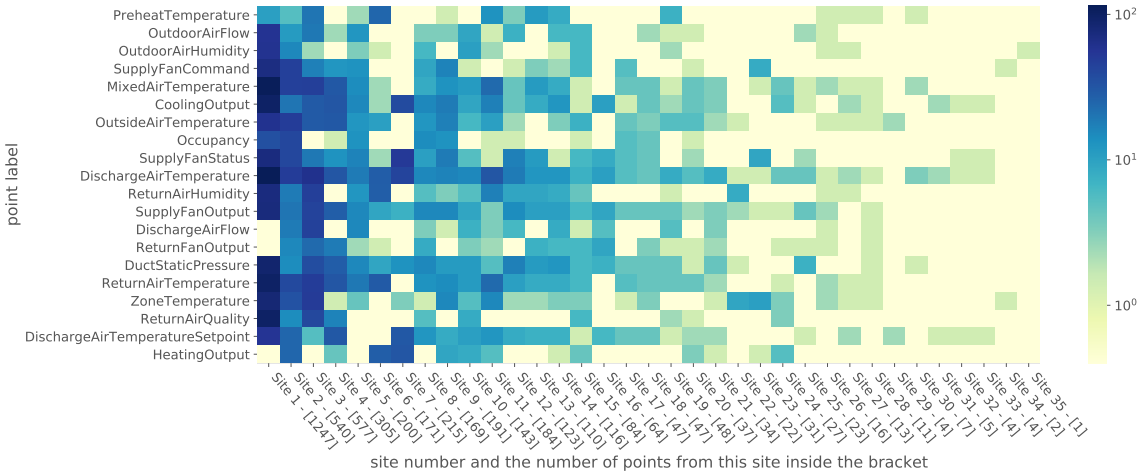


Figure 4.5: Frequency counts of each point label across 35 different sites, the number in the horizontal axis represents the total number of points at this site

4.4 Experiments

We will use the same evaluation strategy S2 as is introduced earlier in Section 3.5.3 to explore the generalizability of the neural network based approaches along with the baseline approaches. That is, we use data from all but one sites to train and use the data from the remaining sites to test, and we iterate over sites.

4.5 Results and Discussions

4.5.1 Metrics

As we have discussed different metrics earlier in Section 3.5.1, we will follow the same rule and use accuracy as the metric in this study. We also provide the detailed performance using other metrics such as F_1 score, precision, recall, and AUC for each class in Appendix C.2.

4.5.2 Comparison of Different Approaches

The average accuracy over different testing sites of the 20-class classification problem can be seen in Table 4.2 for each approach. The two approaches yielding the best score are **winF** and **caeF**, suggesting CNN based approaches can reach the comparable performance with the existing approach.

	statF	winF	tfaF	dtwF	combF	cnn-clf	caeF
accuracy	0.597	0.612	0.446	0.574	0.607	0.565	0.612

Table 4.2: Average accuracy for each approach

To better understand how the accuracy vary when different test sites are being used, we present the accuracy distribution over 35 sites using violin plots in Figure 4.6. As we can see they all have very similar performance near 60%, with **winF** and **caeF** has slightly better results. This confirms that CNN-based unsupervised approach can perform similarly compared with existing statistical-based approaches. It is worth mentioning we have tuned different parameters for both **cnn-clf** and **caeF** approaches and the resulting performance does not change much (less than 3% based on the parameters we explored).

Notice for this 20-class classification problem with unbalanced samples in each class (seen in Figure 4.4), if we have a baseline model predicting every testing

samples to the most frequent class, the resulting accuracy is only 11%, which is much less than 60%. Nevertheless, an accuracy of 60% is not fully reflecting the performance of the model. For **caeF** approach, we do show the other metrics (e.g., precision, recall, F_1 score, AUC) measuring the prediction power for each class in Appendix C.2.

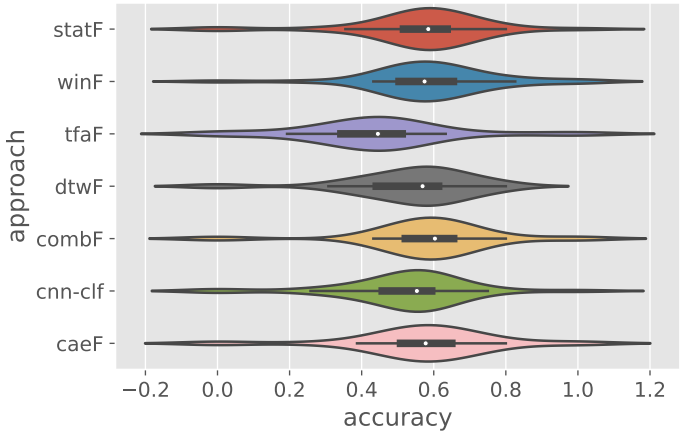


Figure 4.6: Violin plots over 35 sites for different approaches

To visualize the reconstruction capability of CAE, we show an example of how one time series signal can be reconstructed going through convolution, pooling, unpooling, and deconvolution operations in Figure 4.7. The plot under the title is the output of the signal after each operation. The operation after second pooling will produce the latent representation, which is also illustrated in Figure 4.3. The reconstructed signal shown in Figure 4.7 is very close to the input signal, which implicitly suggests the hidden latent representation could be a good approximation (feature) of the original time series.

To further understand how this approach performs for each class, we plot the normalized confusion matrix in Figure 4.8 for **caeF**. As we can see, all temperature sensors, air flow sensors, fan outputs, along with heating and cooling outputs are easily confused. This motivates us to explore the idea of using a hierarchical classifier

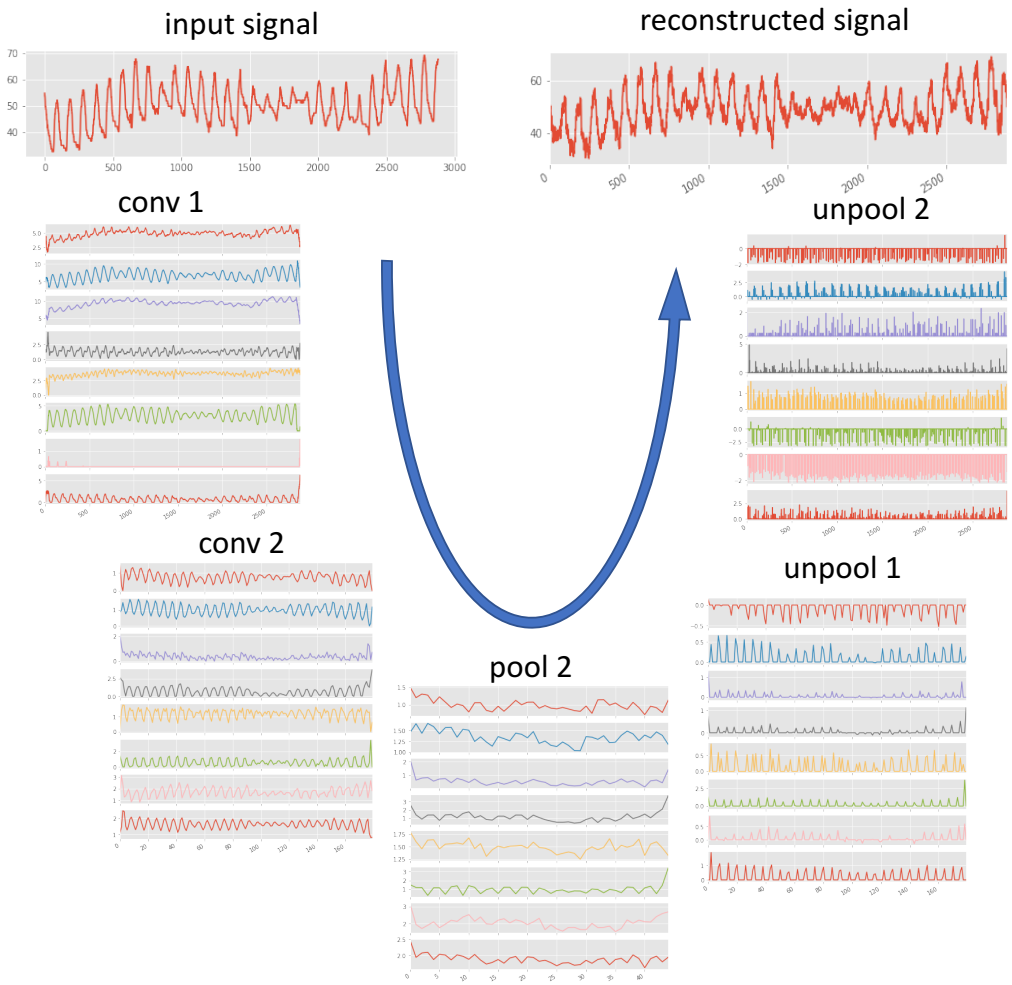


Figure 4.7: An example of how CAE is used to reconstruct the time series signal

where we group 20 different types into larger groups and use two nested classifiers to recognize point types.

4.5.3 A Hierarchical Approach

We define 6 point type groups as follows:

1. Temp: PreheatTemperature, MixedAirTemperature, OutsideAirTemperature, DischargeAirTemperature, ReturnAirTemperature, ZoneTemperature, DischargeAirTemperatureSetpoint

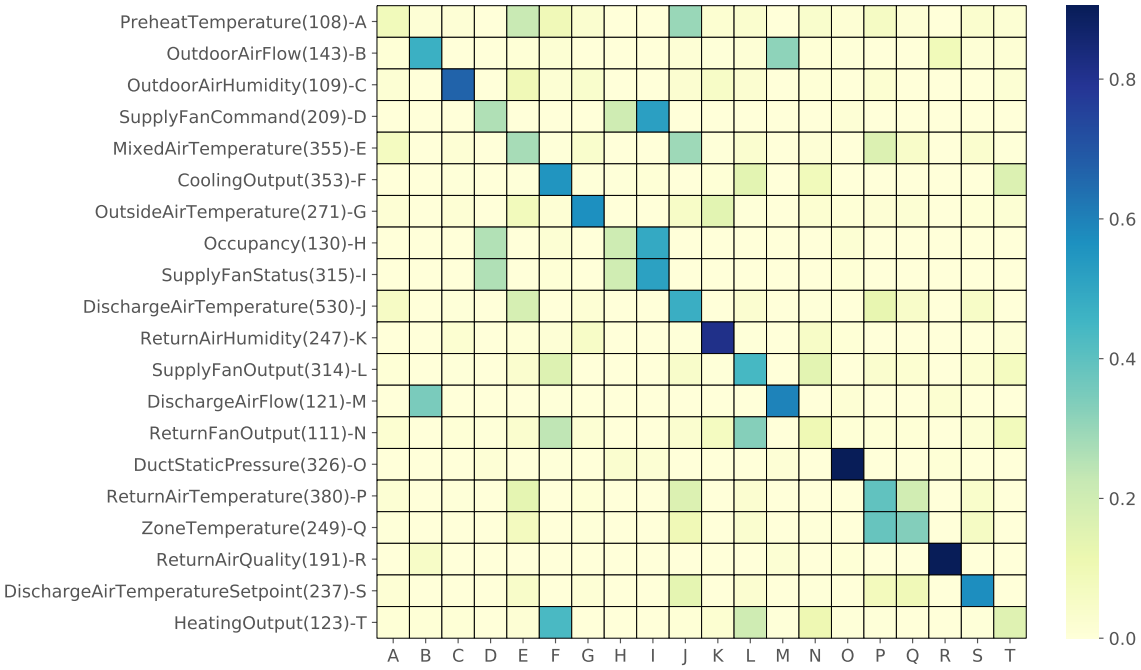


Figure 4.8: Normalized confusion matrix by row using CAE and Random Forest. The number inside the bracket beside the label name on the vertical axis represents the number of testing instances for this class

2. Humidity: OutdoorAirHumidity, ReturnAirHumidity

3. Flow: OutdoorAirFlow, DischargeAirFlow, ReturnAirQuality

4. Output: CoolingOutput, SupplyFanOutput, ReturnFanOutput, HeatingOutput

5. Integer: Occupancy, SupplyFanStatus, SupplyFanCommand

6. Pressure: DuctStaticPressure

With this group, the original 20-class classification task turns into a 6-class classification problem. We can solve this using a high-level classifier. Once the group label is generated, another low-level classifier will be used to produce the specific type of the BAS points. The **caeF** and random forest are used in this study.

To understand the performance of the first high-level classifier, we plot the confusion matrix in log scale shown in Figure 4.9. The number inside corresponds to the count of true positive, false positive, and false negative. The accuracy is calculated by summing over the diagonal and dividing by the total sum of the confusion matrix. As we can clearly see that when the number of classes decreases (the definition of the label changes), the accuracy increases significantly.

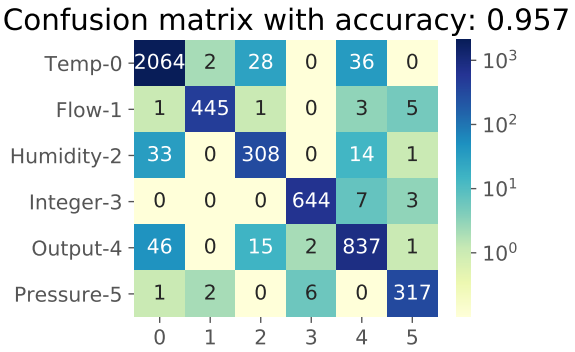


Figure 4.9: Confusion matrix in log-scale using CAE and Random Forest when we group sensors into six groups

Based on such a high accuracy, we proceed to the low-level classier. However, we find that the low-level classifier does not perform well. As a matter of fact, the resulting accuracy quickly drops from 96% down to 61%, which is the same as the one in Table 4.2.

To understand the mechanism of the hierarchical classifier, Figure 4.10 shows the re-ordered confusion matrix in log scale when we group the point types in the same group. As we can see, all temperature sensors are very easily confused with each other. Similarly, the cooling output is also heavily confused with the supply fan output.

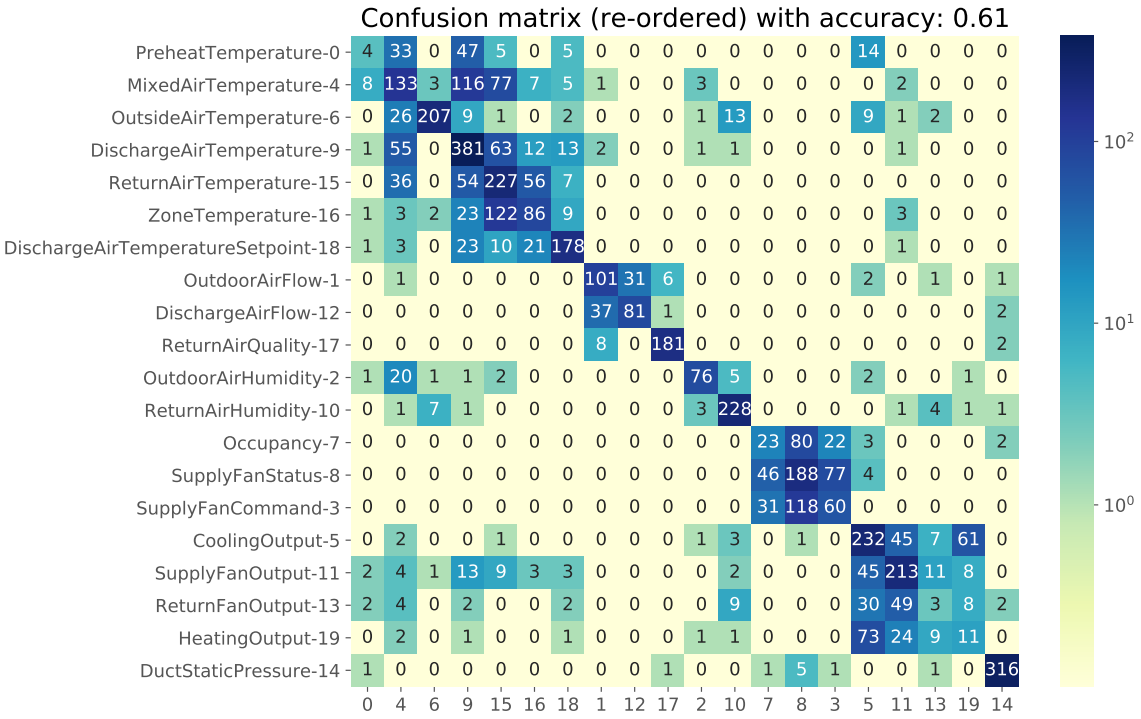


Figure 4.10: Re-ordered confusion matrix in log-scale using CAE and Random Forest

4.5.4 An Ensemble of Classifiers

We notice that different approaches can classify the point types with different levels of accuracy, as is seen in Table 4.3 where the F_1 score for each class is shown for each approach. For example, **caeF** is good at predicting "OutdoorAirHumidity", and **cnn-clf** is slightly better at predicting "Occupancy". We can then use an ensemble approach to combine those two models, which may produce better results than using each alone.

A simple voting strategy based on the probability output of each classifier is used. Specifically, for each prediction, we pick the label assignments that yields the highest probability among all approaches. We find this can help increase the accuracy by up to 3% (from 61% to 64%). For example, including three models **winF**, **cnn-clf** and **caeF** is able to increase the accuracy up to 3%. We also observe that when

	statF	**winF**	**tfaF**	**dtwF**	**combF**	**cnn-clf**	**caeF**
PreheatTemperature	0.07	0.00	0.00	0.00	0.00	0.00	0.03
OutdoorAirFlow	0.50	0.61	0.36	0.56	0.59	0.51	**0.65**
OutdoorAirHumidity	0.78	0.80	0.77	0.78	0.81	0.80	**0.86**
SupplyFanCommand	0.07	**0.12**	0.01	0.05	0.08	0.04	0.11
MixedAirTemperature	0.37	**0.41**	0.14	0.37	0.38	0.26	0.39
CoolingOutput	0.63	**0.69**	0.41	0.58	0.61	0.62	0.66
OutsideAirTemperature	0.76	**0.86**	0.78	0.81	0.79	0.73	0.82
Occupancy	0.46	0.09	0.69	0.73	0.75	**0.82**	0.17
SupplyFanStatus	0.64	0.55	0.62	**0.68**	**0.68**	**0.68**	0.59
DischargeAirTemperature	0.61	**0.64**	0.39	0.54	0.60	0.52	0.62
ReturnAirHumidity	0.84	**0.89**	0.64	0.86	0.85	0.78	**0.89**
SupplyFanOutput	0.64	**0.68**	0.50	0.57	0.61	0.61	0.66
DischargeAirFlow	0.69	0.66	0.47	0.66	**0.70**	**0.70**	0.66
ReturnFanOutput	0.05	0.02	0.00	0.02	0.05	0.03	0.05
DuctStaticPressure	**0.98**	0.97	0.77	0.94	0.97	0.95	0.97
ReturnAirTemperature	**0.51**	**0.51**	0.28	0.44	0.47	0.49	0.48
ZoneTemperature	0.28	0.32	0.18	0.22	0.27	0.14	**0.37**
ReturnAirQuality	0.91	**0.95**	0.62	0.93	0.94	**0.95**	**0.95**
DischargeAirTemperatureSetpoint	0.66	**0.76**	0.39	0.55	0.70	0.34	**0.76**
HeatingOutput	0.13	0.15	0.11	0.01	0.06	0.29	0.08
highlighted counts	2	10	0	1	2	4	6

Table 4.3: Accuracy score for each class and for each approach. The highest score for each class is highlighted in bold if it is higher than 0.5

more models are being used to build the ensemble classifier, the performance will stop increasing and even start decreasing. This is probably due to that certain poor models have higher probability predictions and could win during the voting phase. Hence, the ensemble approach should be used with caution to make sure the right combination of approaches is used.

4.6 Conclusion

In this chapter, we explore a purely data-driven approach based on convolutional neural networks. The approach can generate similar and sometimes better performance than existing approaches. However, when the number of classes increases, the performance rapidly drops. Despite this, we show that we improve the performance slightly (3%) by using ensemble classifiers.

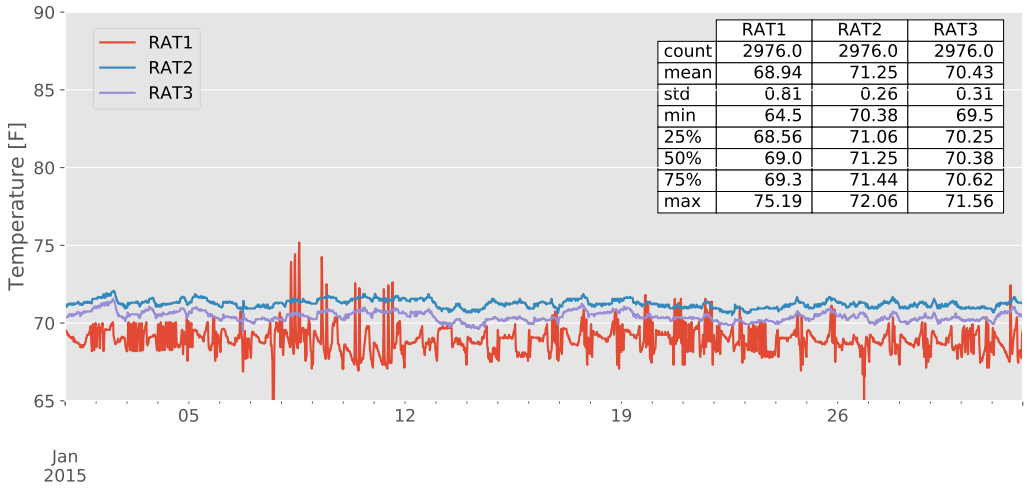
Furthermore, when the model makes incorrect predictions, it is hard to track why the model makes mistakes. There is a need to develop new methods to understand the behaviors of the metadata inference approaches and recognize when the model will fail and why it fails.

Chapter 5

A Physical Model-based Approach

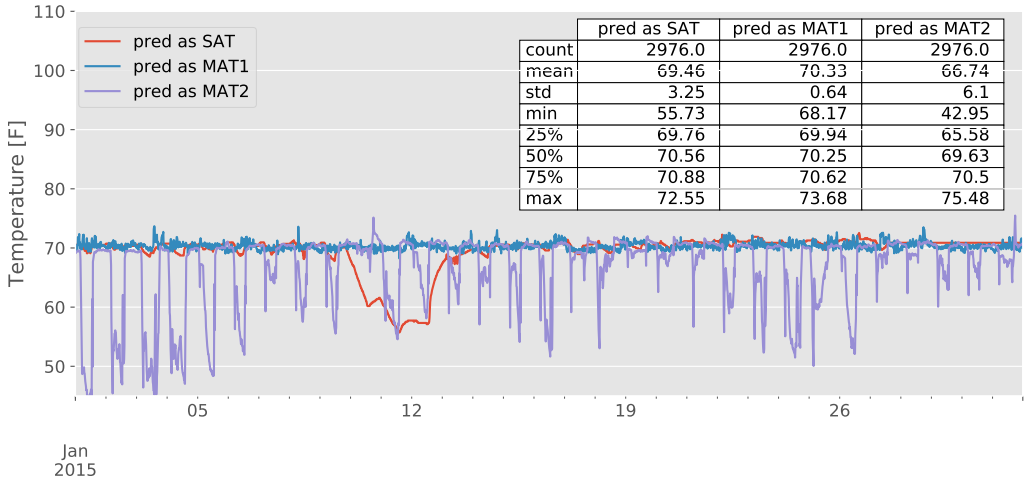
In previous chapters, we have proposed a metadata inference framework and tested it on both single building and multiple buildings using different time series based inference approaches. These approaches rely on extracting statistical properties from the time series, and as such, they are data-driven or black box models. Based on these models, our results show that an expected accuracy ranging from 62% to 75% can be achieved to infer the type information of BAS points required by common FDD approaches. However, when the model makes an incorrect prediction, it is often difficult to explain why it is wrong. The common approach to diagnose these erroneous predictions are to plot the incorrectly labeled time series, calculate statistical quantities, and reason from a statistical perspective.

For example, Figure 5.1 shows the time series data of six return air temperature sensors where three of them are correctly predicted and the remaining three are incorrectly predicted. The summary statistics can be seen in the upper right section of each subplot. We can see that the three incorrectly predicted return air temperature sensors in Figure 5.1b have different patterns in time series data (e.g., a higher standard deviation and wider range) compared with the correctly predicted



	RAT1	RAT2	RAT3
count	2976.0	2976.0	2976.0
mean	68.94	71.25	70.43
std	0.81	0.26	0.31
min	64.5	70.38	69.5
25%	68.56	71.06	70.25
50%	69.0	71.25	70.38
75%	69.3	71.44	70.62
max	75.19	72.06	71.56

(a) Three return air temperature sensors are correctly predicted



	pred as SAT	pred as MAT1	pred as MAT2
count	2976.0	2976.0	2976.0
mean	69.46	70.33	66.74
std	3.25	0.64	6.1
min	55.73	68.17	42.95
25%	69.76	69.94	65.58
50%	70.56	70.25	69.63
75%	70.88	70.62	70.5
max	72.55	73.68	75.48

(b) Three return air temperature sensors are incorrectly predicted as supply air and mix air temperature sensors

Figure 5.1: A total of six time series for return air temperature sensors

ones in Figure 5.1a. The distinct patterns make some return air temperature sensors indistinguishable with supply air and mixed air temperature sensors. However, this interpretation lacks a fundamental understanding of the underlying physical processes, e.g., what are the thermodynamics driving the behaviors of each HVAC system, and might not hold for other time series data from return air temperatures. Hence, a more systematic explanation based on first principles is missing,

and there is a need to incorporate physical models to improve our understanding of the underlying process and further help the task of metadata inference.

Attempts at this have been made previously in [54] where authors utilize an energy estimation model to infer which room the sensor is located. The sensor data are combined with the energy model to understand the thermal performance of different zones inside buildings. The effectiveness of using the physical model to infer the location information motivates us to explore the potential of it further to infer additional metadata such as type information of BAS points. To our knowledge, the physical model-based approach to infer the type information of BAS points has not been studied before.

In this chapter, we hypothesis that with the help of physical models, we can understand the physical process (e.g., how the data values are generated following the law of physics), uncover the underlying relationship of BAS points (e.g., how the BAS points in the same unit can impact each other's data patterns), and better discriminate points that are confused in data-driven models. Explicitly, we define physical models for the mixing box, the cooling coil and the heating coil in an AHU using mathematical equations. Then we propose a new physical model-based approach to recognize BAS points that are easily confused. To validate whether this knowledge from the physical model can help the metadata inference task, we test the approach on both simulation and real-world datasets.

5.1 Physical Model of an AHU

An AHU is an integrated large equipment used to circulate and condition the air being supplied to a building. Its major components include ducts, dampers, a mixing box, fans, cooling, and heating coils as shown in Figure 5.2. Sensors and

actuators in the AHU monitor and change the condition of the unit. A mapping between the acronyms in Figure 5.2 and the full descriptions for these BAS points can be found in Table A.1. A typical way of building a physical model for an AHU is to use a lumped-parameter approach where each component is modeled separately, and then all components are aggregated together, which has been studied as early as in the 1980s [91, 92]. These models include the room and zone model, the heating coil model, the duct and pipe model, the damper model, the valve model, the fan and pump model, the humidifier model, the temperature, and controller models [91], all of which are described by detailed linear and nonlinear differential equations. To provide modeling for chilled water cooling coils, authors in [93] develop a detailed model for a cooling coil based on dynamic forwarding modeling which considers the transient behaviors of the system in addition to the steady states. However, one problem associated with these models is that they are too detailed and require a large number of parameters, making them less practical to be used on site [94]. Meanwhile, researchers in [95, 94, 96] have built simplified models for zones, mixing boxes, heating coils and cooling coils in AHUs. The simplification is achieved with fewer parameters and more assumptions. For example, instead of using an array of temperature sensors to measure the spatial temperature distribution, the simplified model will assume a uniform distribution in space and use only one temperature sensor. Some cooling coil models also consider the humidity changes while others just focus on the temperature and flow rates change.

In addition to modeling AHUs using physical models as we described, there are also black-box data-driven models. As we focus on interpretative models of an AHU in secondary HVAC systems, we will not introduce them in this section. A detailed review of modeling methods for HVAC systems can also be found in [97, 98, 99].

Another field that is related to the HVAC modeling is the virtual sensing technique. It is not a new concept in BAS. The sequences of operations in HVAC rely heavily on the "software points" which are derived points based on the physical existing "hardware points" [22]. For example, the point "enthalpy" is a derived software point based on humidity sensor and temperature sensor to determine the economizer status. These virtual sensors in existing BAS are modeled based on a specific mathematical formula. Although the research in this area is still at a very early stage [100], there are some recent developments on virtual sensing focusing on vapor compression systems [101, 102], chillers [103, 104], heat pumps [105, 101, 102] and AHUs [106, 107].

Specifically, virtual sensors in AHUs include virtual mixed air temperature sensor [106], virtual cooling coil capacity sensor [107], and virtual filter status [108]. These virtual sensing methods are tested either using simulation data or on a small testbed and showed different limitations. For example, [106] proposed a smart mixed-air temperature sensor inside ducts of an AHU, which can provide more accurate results compared with using the single-point measurement of mixed air temperature alone. The increased accuracy is achieved through combining information from damper control signal, outdoor and return air temperature. However, such method still needs at least one mixed air temperature sensor to reduce the error and is mainly applicable to constant-air-volume (CAV) systems. [107] developed models for direct expansion (DX) coils using manufactured data to generate virtual cooling coil capacity sensor.

It is worth pointing out that all physical models are approximations to the dynamics of the actual system based on mathematical equations that represent an abstraction of their physical behaviors. The comprehensive and complicated models

can simulate the systems with more details while it could be difficult in practice to properly configure and commission them [109]. In our case, we are not interested in a particular single AHU to model with sufficient details and requiring configurations and commissions; instead, we are more interested in the general physical behaviors of AHU components which can be modeled using sensors already instrumented inside BASs. Hence, we select simplified models for three AHU components we are interested in.

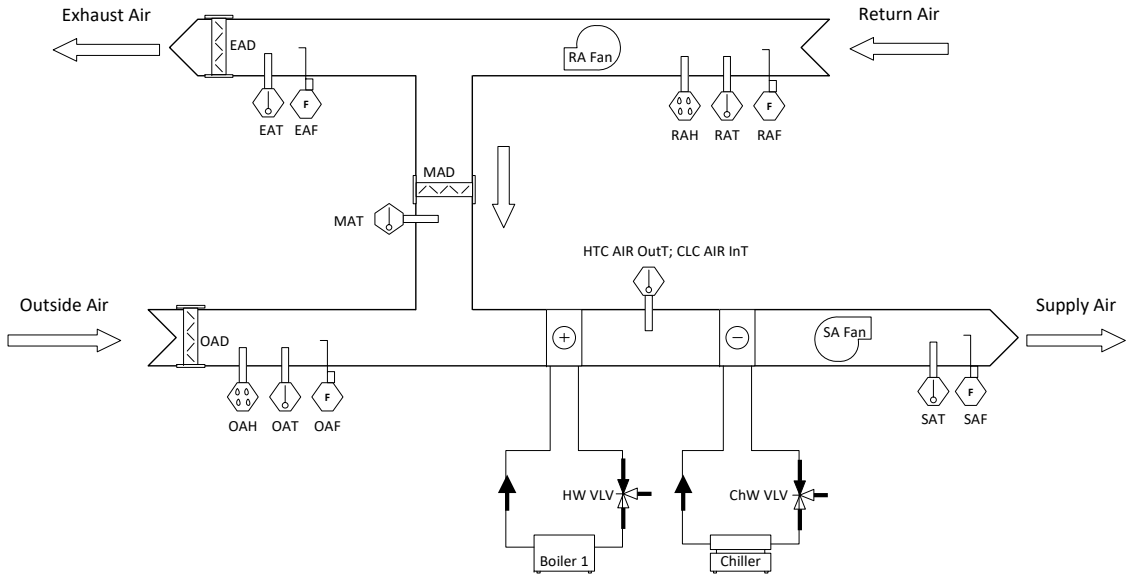


Figure 5.2: Schematic diagram of a typical AHU

In this section, we specifically focus on three major components in AHU, namely, the mixing box, the cooling coil, and the heating coil. The variables and parameters that are used in the models throughout this chapter can be seen in Table 5.1. It is worth mentioning that the flow rate can be measured using either mass flow rate $\dot{m}$ [kg/s] or volume flow rate $\dot{V}$ [m^3/s]. They can be easily converted by multiplying the density of the medium (air or water). For consistency, we use the mass flow rate in all of our models.

Symbol	Unit	Description
C_c	kJ/(kg°C)	specific heat of the cooling coil
C_h	kJ/(kg°C)	specific heat of the heating coil
C_{pa}	kJ/(kg°C)	specific heat of air
C_{pw}	kJ/(kg°C)	specific heat of water
f_{out}	%	outdoor air fraction
$\dot{m}_{out}$	kg/s	outdoor air flow rate
$\dot{m}_{sup}$	kg/s	supply air flow rate
$\dot{m}_{w,c}$	kg/s	chilled water flow rate
$\dot{m}_{w,c}^{max}$	kg/s	maximum chilled water flow rate
$\dot{m}_{w,h}$	kg/s	hot water flow rate
$\dot{m}_{w,h}^{max}$	kg/s	maximum hot water flow rate
M_c	kg	air quality of cooling coil
M_h	kg	air quality of heating coil
T_{mix}	°C	mix air temperature
T_{out}	°C	outside air temperature
T_{ret}	°C	return air temperature
T_{sup}	°C	supply air temperature
T_c	°C	cooling coil outlet air temperature
$T_{ai,c}$	°C	cooling coil inlet air temperature
$T_{wi,c}$	°C	cooling coil inlet water temperature
$T_{wo,c}$	°C	cooling coil outlet water temperature
T_h	°C	heating coil outlet air temperature
$T_{ai,h}$	°C	heating coil inlet air temperature
$T_{wi,h}$	°C	heating coil inlet water temperature
$T_{wo,h}$	°C	heating coil outlet water temperature
u_c	%	percentage of cooling coil valve position
u_h	%	percentage of heating coil valve position

Table 5.1: Nomenclature table

5.1.1 Mixing Box

A mixing box is the section of an AHU to mix the return air with the outside air through a mixed air damper controlling the mixing fraction, as is shown in Figure 5.2. This mixed air damper also serves as a device to optimize energy usage by making use of the heat from the return air in winter when the return air temperature is higher than the outside air and the cooling capacity of it in summer when the

return air temperature is lower than the outside air. Another outdoor air damper is used to control the outdoor air fraction. Once the air is mixed, it is conditioned through heating and cooling coils to be supplied to zones. Modeling methods for the mixing box have been proposed in both [95, 94, 110] where relationships between temperature measurements and air flow rates are defined. These models are used to study optimal control strategies and automated fault diagnostics, based on energy and mass balance equations.

Though the two models have similar objectives and complexity, the one in [95] requires the mixed air flow rate as an input, and this measurement is often unavailable in current building systems as is seen earlier in Section 2.3. Hence, we adopt the model from [94] where we use outdoor air flow rate, supply air flow rate, return air temperature, mixed air temperature, and outdoor air temperature to build the physical model. We start by defining the outdoor air fraction as the ratio between the "supply air mass flow rate" and "the outdoor air flow rate":

$$f_{out} = \frac{\dot{m}_{out}}{\dot{m}_{sup}} \tag{5.1}$$

Since there is a portion of return air mixing with the outside air to produce the supply air, f_{out} represents the percentage of the outdoor air during the mixing process. On the other hand, this fraction is also related to the temperature difference between the return air and both the mixed air and the outdoor air. In other words, it is also the ratio between the temperature difference before the mixing box, and after the mixing box:

$$f_{out} = \frac{T_{ret} - T_{mix}}{T_{ret} - T_{out}} \tag{5.2}$$

The equation can be rewritten as:

$$T_{mix} = f_{out}T_{out} + (1 - f_{out})T_{ret} \tag{5.3}$$

$$= \frac{\dot{m}_{out}}{\dot{m}_{sup}}T_{out} + (1 - \frac{\dot{m}_{out}}{\dot{m}_{sup}})T_{ret} \tag{5.4}$$

which suggests the mixed air temperature is a linear combination of outdoor air temperature and return air temperature weighted by the outdoor air fraction. As a result, we have built a mathematical relationship between these five sensors. Given inputs from four sensors, we can estimate the remaining one. Notice this model is an approximation but it is able to capture the underlying physical process, as it has been shown in previous publications [111, 94, 110].

5.1.2 Cooling Coil

Both cooling coils and heating coils are heat exchangers which transfer the energy between the fluid and the air. As is seen in Figure 5.2, the cooling coil circulates the chilled water from the chiller to cool down the air passing through the coil. The amount of cooling capacity is controlled using a chilled water valve regulating the flow rate of the water. Comprehensive modeling of the cooling coil has been previously seen in [93] where authors use a dynamic modeling approach considering the transient behaviors of the cooling coil with sufficient details (e.g., modeling the spatial and time distributions of the temperature and humidity of the coil) and [112] where researchers build a coupled model of the cooling coils together with the temperature sensors inside VAV boxes. Nevertheless, these two models are difficult to be used in our study given that many inputs and parameters are required by the models. In [95], an empirical model is adopted to approximate the

air temperature using the water temperature as follows:

$$T_c = 0.0587T_{wo,c}^2 + 1.773T_{wo,c} + 1.1816 \tag{5.5}$$

However, such an equation lacks sufficient physical interpretations and makes the assumption that air temperature is only determined by water temperature using a second-order polynomial equation with coefficients being fixed constants. Such an assumption is less likely to generalize to cooling coils in different AHUs. In [96] researchers use a model which is both simplified and have physical foundations based on the law of conservation of energy. Hence, given the reasons above, we will use this model in our study. Following the definition in [96], the thermal balance is defined as:

$$M_c C_c \frac{dT_c}{dt} = \dot{m}_{sup} C_{pa}(T_{ai,c} - T_c) - \dot{m}_{w,c} C_{pw}(T_{wo,c} - T_{wi,c}) \tag{5.6}$$

$\dot{m}_{w,c}$ can be represented in terms of u_c [94]:

$$\dot{m}_{w,c} = \dot{m}_{w,c}^{max} \cdot {u_c}^2$$

In Equation 5.6, the parameters with fixed values include $M_c, C_c, C_{pa}, C_{pw}, \dot{m}_{w,c}^{max}$, the variables include $T_c, T_{ai,c}, T_{wo,c}, T_{wi,c}, \dot{m}_{sup}, u_c$. If we assume $T_{ai,c}, T_{wo,c}, T_{wi,c}, \dot{m}_{sup}, u_c$ do not change the values in a short time period from t_0 to t_1, Equation 5.6 can be approximately solved:

$$\frac{dT_c}{dt} = -\frac{\dot{m}_{sup}C_{pa}}{M_cC_c}T_c + \frac{\dot{m}_{sup}C_{pa}T_{ai,c} - \dot{m}_{w,c}C_{pw}(T_{wo,c} - T_{wi,c})}{M_cC_c} \tag{5.7}$$

$$\frac{dT_c}{dt} = a_cT_c + b_c \tag{5.8}$$

$$T_c^{t_1} = \frac{(a_cT_c^{t_0} + b_c)\exp^{a_c(t_1-t_0)} - b_c}{a_c} \tag{5.9}$$

where $a_c = -\frac{\dot{m}_{sup}C_{pa}}{M_cC_c}$, $b_c = \frac{\dot{m}_{sup}C_{pa}T_{ai,c} - \dot{m}_{w,c}C_{pw}(T_{wo,c}-T_{wi,c})}{M_cC_c}$.

Given the values from the sensors $T_{ai,c}, T_{wo,c}, T_{wi,c}, \dot{m}_{sup}, u_c$ and an initial value $T_c^{t_0}$, we are able to model T_c over time.

5.1.3 Heating Coil

As early as in the 1980s, a very detailed modeling method for heating coils was presented in [91] where the relationships between temperatures, air pressures, frictional coefficients and other specific variables in the coil are modeled based on the number of transfer units (NTU) methods. The purpose of the simulation model is to study fault diagnosis and energy optimization. Another heating coil model in [95] is later proposed to study various control strategies to improve the efficiency. This model requires the humidity measurements of the air before and after the coils for the mass balance equation. However, both models require variables which are not commonly seen in existing buildings, as is seen in Section 2.3 where the common BAS points in AHUs are listed. To have a model that is both simple and has physical interpretations, we adopt the same heater exchanger model presented in [96]. The heating coil is similar to how we model the cooling coil, which is defined as:

$$M_hC_h\frac{dT_h}{dt} = \dot{m}_{sup}C_{pa}(T_{ai,h} - T_h) - \dot{m}_{w,h}C_{pw}(T_{wo,h} - T_{wi,h}) \tag{5.10}$$

And we can solve T_h in a similar manner as

$$T_h^{t_1} = \frac{(a_h T_h^{t_0} + b_h)e^{a(t_1 - t_0)} - b_h}{a_h} \tag{5.11}$$

where $a_h = -\frac{\dot{m}_{sup} C_{pa}}{M_h C_h}$, $b_h = \frac{\dot{m}_{sup} C_{pa} T_{ai,h} - \dot{m}_{w,h} C_{pw} (T_{wo,h} - T_{wi,h})}{M_h C_h}$.

5.2 Model-based Metadata Inference Approach

In this section, we propose a model-based approach to infer the type of BAS points. The underlying reasoning is that if we can find the label assignment (prediction of the point types) that fits the model best, this assignment will most likely be the correct assignment. This is related to the problem of system identification where the mathematical models of dynamical systems are built based on observed data from the systems [113]. However, instead of collecting data and picking the "best" model from the model set based on data, we approach it the other way around assuming we have known the model and we want to pick the "best" data with the correct label assignment that fits the model. We describe this approach in detail as follows.

Denote $\mathcal{F}$ as a set of models describing the physical process of the HVAC system. For any model $f \in \mathcal{F}$, we can implicitly define the model using the equation:

$$f(\mathbf{X}, \boldsymbol{\theta}) = 0 \tag{5.12}$$

where $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ represents n variables used in the model and $\boldsymbol{\theta}$ represents the relevant parameters. These n variables are distinct BAS points which can be

labeled as

$$L_X = (1, 2, \cdots, n) \tag{5.13}$$

Without loss of generality, we can represent the model explicitly as

$$\mathbf{x}_1 = g(\mathbf{X}_{\setminus \mathbf{x}_1}, \boldsymbol{\theta}) \tag{5.14}$$

where $\mathbf{X}_{\setminus \mathbf{x}_1} = (\mathbf{x}_2, \cdots, \mathbf{x}_n)$.

Now, given m time series $\mathbf{Z} = (\mathbf{z}_1, \cdots, \mathbf{z}_m)$ from m BAS points in one unit[1], we want to generate the labels for m time series, i.e., find which BAS points correspond to these m time series. It is required that $m \geq n$ meaning the set $\mathbf{Z}$ should at least include all the variables (BAS points) required in model f. We denote the label set $L_Z = (L_{z_1}, L_{z_2}, \cdots, L_{z_m})$ where one example could be

$$L_Z = (1, 2, \cdots, n, 0, \cdots, 0) \tag{5.15}$$

representing the first n time series in $\mathbf{Z}$ correspond to the labels $(1, 2, \cdots, n)$, and the remaining $(m - n)$ time series marked as 0 does not belong to any BAS points in the label set L_X for this model. It is worth mentioning that, because we assume m BAS points are from the same unit and each point will have a different label, the true label set L_Z^* for these m time series will be a re-ordered version of L_Z, or a permutation of the original sequence.

Our goal is to find the true label set L_Z^* for time series data $\mathbf{Z}$. We argue that if the time series data with correct labels are identified, the model should fit the

[1]Here we assume we have identified BAS points that are in the same unit (e.g. AHU) but have not recognized the identity of each point. We further assume that there are no redundant BAS points in the unit and each point will have a different label.

data "better" compared to the data with incorrect labels. Notice we do not fit data to models, but rather the other way around by evaluating how well the model can explain the data. The goodness of fit can be quantified using a fitness function. Suppose we have found one set of labels $L_Z = (L_{z_1}, L_{z_2}, \cdots, L_{z_m})$. We can extract n time series $\tilde{\mathbf{Z}}$ required by the model as

$$\tilde{\mathbf{Z}}(L_Z) = (\tilde{\mathbf{z}}_1, \cdots, \tilde{\mathbf{z}}_n) \tag{5.16}$$

$$= \left(\sum_{i=1}^{m} \mathbf{z}_i \mathbb{1}(L_{z_i} = 1), \cdots, \sum_{i=1}^{m} \mathbf{z}_i \mathbb{1}(L_{z_i} = n) \right) \tag{5.17}$$

which is a subset of permuted $\mathbf{Z}$.

Notice that $\tilde{\mathbf{Z}}$ is determined by the label set L_Z. Using Equation 5.14, we can predict $\tilde{\mathbf{z}}_1$ as

$$\hat{\mathbf{z}}_1 = g(\tilde{\mathbf{Z}}(L_Z)_{\setminus \tilde{\mathbf{z}}_1}, \boldsymbol{\theta}) \tag{5.18}$$

If the label set L_Z correctly depicts the true labels for $\mathbf{Z}$, then $\hat{\mathbf{z}}_1$ should be very close to $\tilde{\mathbf{z}}_1$. We use a fitness function $h(\tilde{\mathbf{z}}_1, \hat{\mathbf{z}}_1)$ to evaluate how close they are (e.g., the goodness of fit). If a large value of h indicates more similarities, the problem is converted to finding

$$\hat{L}_Z^* = \arg\max_{L_Z} h\left(\tilde{\mathbf{z}}_1(L_Z), \hat{\mathbf{z}}_1(L_Z)\right) \tag{5.19}$$

by iterating over all possible permutations L_Z.

Theoretically, we need to iterate $\frac{m!}{(m-n)!}$ times as there are $\frac{m!}{(m-n)!}$ different ways to arrange label sequences (label assignments) for L_Z; in practice, we might know

which permutations are infeasible, so we can rule out some sequences. For example, if we know the true label for $\mathbf{z}_1$ is 1, all label assignments that are not labeling $\mathbf{z}_1$ as 1 can be eliminated. Hence, we can focus on the permutations which have the labels for points we are uncertain about.

An example from the mixing box

To better understand the model-based metadata inference approach, we provide a simple example based on the mixing box. Here a mapping between the components of the approach and the model can be seen in Table 5.2. The model f and variables $\mathbf{X}$ are explained in the first two rows. Notice that this model is quite simple, and we do not have parameters $\boldsymbol{\theta}$ associated it with. The corresponding label set and the names for the variables $\mathbf{X}$ are specified. We further convert the model from the implicit form to the explicit form regarding one variable $\mathbf{x}_5$.

Approach	Mixing Box Model
$f(\mathbf{X}, \boldsymbol{\theta}) = 0$	$\frac{T_{ret} - T_{mix}}{T_{ret} - T_{out}} - \frac{\dot{m}_{out}}{\dot{m}_{sup}} = 0$
$\mathbf{X}$	$(\dot{m}_{out}, \dot{m}_{sup}, T_{out}, T_{ret}, T_{mix})$
$\boldsymbol{\theta}$	ϕ
L_X	$(1, 2, 3, 4, 5)$
L_X name	(OAF, SAF, OAT, RAT, MAT)
$\mathbf{x}_5$	T_{mix}
$\mathbf{x}_5 = g(\mathbf{X}_{\setminus \mathbf{x}_5}, \boldsymbol{\theta})$	$T_{mix} = \frac{\dot{m}_{out}}{\dot{m}_{sup}} T_{out} + (1 - \frac{\dot{m}_{out}}{\dot{m}_{sup}}) T_{ret}$
$\mathbf{Z}$	$(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5)$
L_Z	$(?, ?, ?, ?, ?)$

Table 5.2: A mapping between the element in the approach and the mixing box model

Now, given time series data $\mathbf{Z}$ from five sensors shown in Figure 5.3, we need to find the corresponding true label set L_Z for these five sensors. To achieve that, suppose we generate two possible label assignments $(1, 2, 3, 4, 5)$ and $(1, 2, 3, 5, 4)$ as is seen in Table 5.3. If we can find which one of these two has a higher probability of

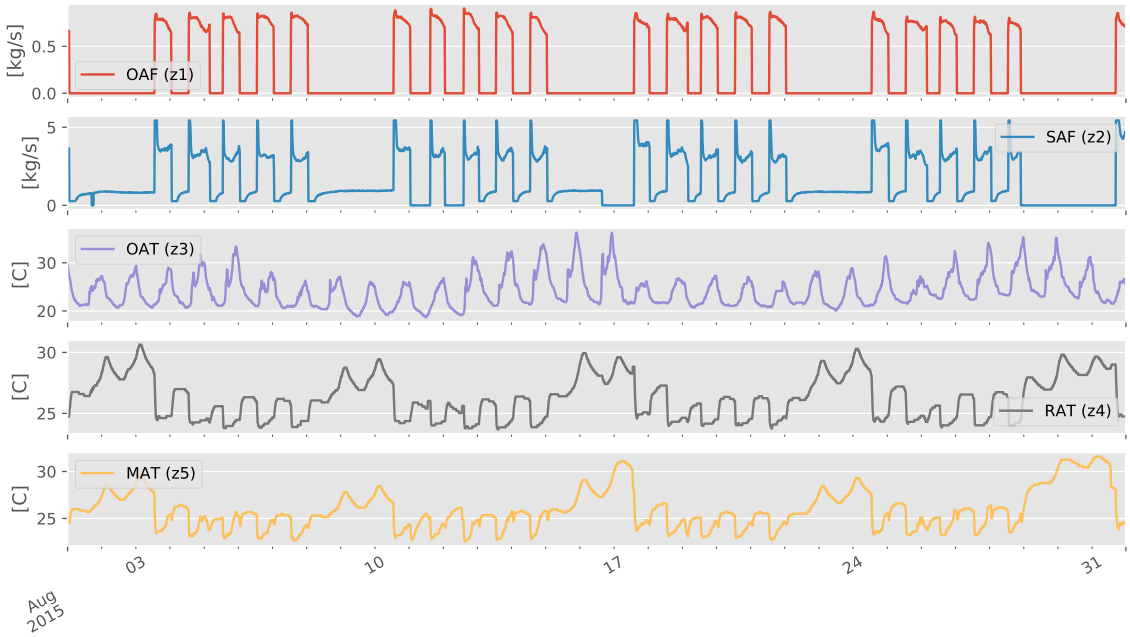


Figure 5.3: An example of raw time series plots from five sensors in the mixing box

being the correct label set, then we can theoretically iterate all possible permutations (a total of $5! = 120$ permutations) and identify the most probable correct assignment. For illustration purpose, let us assume we are certain about $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ being (1-OAF, 2-SAF, 3-OAT), and we do not know the label assignments for $\mathbf{z}_4, \mathbf{z}_5$. We evaluate both assignments and use the fitness function h to find out which one is more probable. The third row in Table 5.3 is the process of extracting the necessary BAS points for the model-based on the label assignment L_Z. The fourth row is to predict $\hat{\mathbf{z}}_5$ using the rest variables following Equation 5.3. Then we evaluate how close is the predicted variable $\hat{\mathbf{z}}_5$ to the true value $\tilde{\mathbf{z}}_5$ using a fitness function h.

In this case, we use R^2 score as the fitness function, which is also known as the coefficient of determination and measures the proportion of the total variance of the dependent variable (true value) explained by the model (predicted value). Denote

Label Assignment 1	Label Assignment 2
$L_Z^{(1)} = (1, 2, 3, 4, 5)$	$L_Z^{(2)} = (1, 2, 3, 5, 4)$
$\tilde{\mathbf{Z}}(L_Z^{(1)}) = (\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2, \tilde{\mathbf{z}}_3, \tilde{\mathbf{z}}_4, \tilde{\mathbf{z}}_5) = (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5)$	$\tilde{\mathbf{Z}}(L_Z^{(2)}) = (\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2, \tilde{\mathbf{z}}_3, \tilde{\mathbf{z}}_4, \tilde{\mathbf{z}}_5) = (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_5, \mathbf{z}_4)$
$\hat{\mathbf{z}}_5(L_Z^{(1)}) = \frac{\tilde{\mathbf{z}}_1}{\tilde{\mathbf{z}}_2}\tilde{\mathbf{z}}_3 + (1 - \frac{\tilde{\mathbf{z}}_1}{\tilde{\mathbf{z}}_2})\tilde{\mathbf{z}}_4$	$\hat{\mathbf{z}}_5(L_Z^{(2)}) = \frac{\tilde{\mathbf{z}}_1}{\tilde{\mathbf{z}}_2}\tilde{\mathbf{z}}_3 + (1 - \frac{\tilde{\mathbf{z}}_1}{\tilde{\mathbf{z}}_2})\tilde{\mathbf{z}}_4$
$h\left(\tilde{\mathbf{z}}_5(L_Z^{(1)}), \hat{\mathbf{z}}_5(L_Z^{(1)})\right) = 0.80$	$h\left(\tilde{\mathbf{z}}_5(L_Z^{(2)}), \hat{\mathbf{z}}_5(L_Z^{(2)})\right) = 0.62$

Table 5.3: An evaluation process for two label assignments

$\tilde{\mathbf{z}}$ as the true value and $\hat{\mathbf{z}}$ as the predicted value, it is defined as:

$$h_{R^2}(\tilde{\mathbf{z}}, \hat{\mathbf{z}}) = 1 - \frac{||\tilde{\mathbf{z}} - \hat{\mathbf{z}}||^2}{||\tilde{\mathbf{z}} - \bar{z}||^2} = 1 - \frac{\sum_i (\tilde{z}_i - \hat{z}_i)^2}{\sum_i (\tilde{z}_i - \bar{z})^2} \tag{5.20}$$

where $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} \tilde{z}_i$.

The R^2 score indicates how well the model fits the data (the higher, the better). As we can see in the last row of Table 5.3, the label assignment 1 has a higher value compared with label assignment 2. Hence, it is more likely the label assignment 1 is correct, which is aligned with the ground truth where we use "RAT" ($\mathbf{z}_4$) to predict "MAT" ($\mathbf{z}_5$). Meanwhile, in assignment 2 we use "MAT" ($\mathbf{z}_5$) to predict "RAT" ($\mathbf{z}_4$)., which is not aligned with the model and gives us a smaller R^2 score. The predicted results from two label assignments can be seen in Figure 5.4. If the prediction is perfect, we should see a straight line $y = x$. However, due to many impacting factors (e.g., data collection errors, noise and disturbances to the sensor hardware, the limitations of the model, etc.), we are only seeing a linear trend indicating the model approximately captures the behavior of these sensors.

As we have demonstrated that the proposed physical model-based inference approach can classify BAS points using a simple example, we now proceed to validate the hypothesis that the approach can help the inference task on larger datasets.

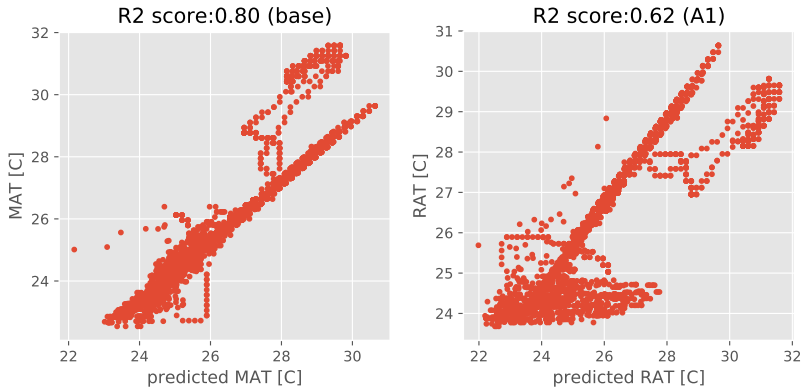


Figure 5.4: An example showing how MAT is recognized against RAT, the left figure shows the R^2 score of actual mix air temperature versus predicated values, the right figure shows R^2 score of actual return air temperature versus incorrectly predicated values

5.3 Data

Attempting to test the hypothesis using real-world time series data directly could be challenging as we have limited control of the process that generates the data. Various issues could occur during the data collection process, including but not limited to gaps in the data, out of range readings, inconsistent sampling rates, sensor faults leading to biased values, noisy external conditions, etc. There are remedies to some of the issues (e.g., outliers could be corrected based on statistical methods and domain knowledge; downsampling or upsampling could be applied to convert the sampling rate of the data, etc.). However, these solutions are providing approximations to the original time series sequence and might distort the original data reflecting the physical process. Additionally, many other issues associated with the real-world data remain to be addressed.

To rule out such variation due to the quality of data and focus on the underlying physical process, we study how the proposed model-based metadata inference approach performs on simulation data first. Then, we further evaluate the approach on real-world datasets. In this section, we begin by describing both of these datasets.

5.3.1 Simulation Data

A simulation tool will be used to generate synthetic measurement data for the variables used in the physical models. There are several software packages available for HVAC simulations including EnergyPlus [114], MATLAB-Simulink [2], HVACSIM+ [115], TRNSYS [116], and Dymola [117]. All these software packages use differential equations to describe the underlying thermodynamic processes of the HVAC system. The differences between these packages are the scope and the details they include in the model for each component. For example, EnergyPlus concentrates on modeling energy consumption and water use in buildings and can generate the energy usage patterns of HVAC components while HVACSIM+, TRNSYS, and Dymola focus on the thermal dynamics perspectives of the modeling considering heat balance, mass balance and flow balance. In our study, we are more interested in the thermodynamics of the system, and Dymola is used. Dymola has gained popularity in the simulation community as it is based on the simulation language Modelica [118] and has been widely used to model complex dynamic systems. There are also many open-source Modelica libraries available as well, e.g., Buildings library maintained by LBL [119].

In our particular simulation model, a 12-story large office building with the floor area of 498,588 ft^2 is used as the testbed, which is one of DOE reference buildings [120]. The model has been tunned following the standards and building codes required by the city of Chicago. This model was run for one week under three different weather conditions including spring, summer, and winter. The simulation was repeated with and without control strategies applied to the building, which eventually produces results of six runs where each run corresponds to one-week long

[2] `https://www.mathworks.com/products/simulink.html`

simulation for the building. Each building has three AHUs located in the bottom, the middle and the top section of the building. In total, we have 18 instances that can be used for our study. The simulations are internally run at the one-second resolution, and the data are only recorded when there is a change. We then sample all the data to a one-minute resolution to conduct the analysis.

Table 5.4 summarizes the simulation dataset to be used including the BAS points required in each model of the AHU. In addition to the points initially required by three models (mixing box, cooling, and heating coil), we also include the BAS points that are easily confused using data-driven models, as is seen from the confusion matrix in Figure 4.10. For example, "SAT" and "RAT" are often confused; "HW VLV / ChW VLV" and "SF SPEED" are often confused. We include the additional points to find out whether the physical model-based approach can discriminate the confused pairs.

Model	**# of instances**	**BAS points used in each AHU**
Mixing box	18	OAF, SAF, SAT, MAT, OAT, RAT
Cooling coil	18	CLC AIR InT, CLC AIR OutT, CLC WATER InT, CLC WATER OutT, ChW VLV, SAF, SF SPEED
Heating coil	18	HTC AIR OutT, HTC WATER InT, HTC WATER OutT, HW VLV, ChW VLV, SAF, SF SPEED

Table 5.4: A summary of the simulation dataset to be used

5.3.2 Real-world Data

For our real-world testbed, We use the same dataset that is described in Section 3.3. However, to make sure one unit contains all the BAS points required by the models, we have to eliminate many AHUs from our original dataset. For the mixing box, we can extract 32 AHUs that include all five sensors required by the

mixing box model. For the cooling coil and heating coil, it is not common for the real buildings to record the air temperature before and after the coil, especially the air temperature before the cooling coil and after the heating coil. As a result, we relax the condition when selecting AHUs. We keep the AHU as long as it has all the required points by the model except for the air temperature before and after the coil. Since the model requires the air temperature measurements around the coil, we will use other temperature sensors to approximate the measurements, which will be discussed in the next section. Additionally, to make sure we have enough testing examples, we divide one-year-long data into 12 months and treat them as 12 different instances.

Table 5.5 summarizes the real-world dataset to be used including the BAS points required in each model of the AHU. Similar to the simulation data, we also include additional measurements that are easily confused in the data-driven models.

Model	**# of instances**	**BAS points used in each AHU**
Mixing box	384	OAF, SAF, SAT, MAT, OAT, RAT
Cooling coil	96	MAT, SAT, CLC WATER InT, CLC WATER OutT, ChW VLV, SAF, SF SPEED
Heating coil	12	MAT, SAT, HTC WATER InT, HTC WATER OutT, HW VLV, ChW VLV, SAF, SF SPEED

Table 5.5: A summary of the real-world dataset to be used

5.4 Experiments

In this section, we describe the experiments conducted to evaluate the capabilities of physical model-based approaches. We specifically explore how the approach can help classify the confused point types from the confusion matrix in Figure 4.10. Similar to the example provided earlier in Section 5.1, we assume we have recognized

the BAS points that belong to the same unit, and we know the labels (the point type) for some of the points. We want to infer the type information of the remaining BAS points that have not been labeled. To achieve that, we assign possible labels to these BAS points. For each assignment, we apply the model to the data to generate the predicted values based on the labels. Then we evaluate the goodness-of-fit of the model-based on a fitness function by comparing the predicted values with the true values. The label assignment that gives the best fit will be the predicted label for these BAS points.

In addition to R^2 score used in the simple example in Section 5.1, other fitness functions h will be used as well. Jf we denote the predicted variable as $\hat{\mathbf{z}}$ and the true value as $\tilde{\mathbf{z}}$ ($\hat{\mathbf{z}}, \tilde{\mathbf{z}} \in \mathbb{R}^n$), then we have the following fitness functions:

1. **DTW**: Dynamic time warping for measuring similarity between two temporal sequences [121] is used, and the Euclidian distance measure is selected to find the warping path.
2. **MAE**: Mean absolute error loss defined as $h_{MAE} = \frac{1}{n}\sum_i |\tilde{z}_i - \hat{z}_i|$.
3. **MSE**: Mean squared error loss defined as $h_{MSE} = \frac{1}{n}\sum_i (\tilde{z}_i - \hat{z}_i)^2$.
4. **MedAE**: Median absolute error loss defined as $h_{MedAE} = median(|\tilde{z}_1 - \hat{z}_1|, \cdots, |\tilde{z}_n - \hat{z}_n|)$.

Notice that these four fitness functions will produce a smaller value if the model fits the data well (the predicted values are close to the true values). Hence **arg min** will be used to select the best label assignments instead of **arg max**.

Armed with these metrics, we now describe the experiments on mixing boxes, cooling coils and heating coils to recognized different point types.

1. **Mixing Box**: classify different temperature sensors

   In the mixing box model, we assume we have recognized "SAF, OAF, OAT" but not "MAT" or "RAT". In other words, for each unit i, we have two time series $\mathbf{z}_1^{(i)}$ and $\mathbf{z}_2^{(i)}$ and we want to find out which one is "MAT" and which one is "RAT". First, we assume $\mathbf{z}_1^{(i)}$ is "MAT" and $\mathbf{z}_2^{(i)}$ is "RAT", and we use the model to produce the predicted "MAT" using $\mathbf{z}_2^{(i)}$ following Equation 5.3. By comparing predicted "MAT" with $\mathbf{z}_1^{(i)}$ using the fitness function, we can get a scalar measuring the goodness-of-fit of this assignment. Now we assume $\mathbf{z}_1^{(i)}$ is "RAT" and $\mathbf{z}_2^{(i)}$ is "MAT", and we predict "MAT" using $\mathbf{z}_1^{(i)}$ and compare it with $\mathbf{z}_2^{(i)}$ using the same fitness function to get another scalar. By comparing these two scalars, we can conclude which label assignment is more probable. We repeat this step for a total of M units (the second column in Table 5.4 and Table 5.5) and count the number of units where we can generate the correct assignment. The performance is reported using accuracy.

   In addition to recognizing the pair "MAT" versus "RAT", we also recognize the pair "RAT" and "SAT" following the same logic. The difference is that we assume we know the labels for "SAF, OAF, OAT, MAT' but not for "RAT" or "SAT".

2. **Cooling Coil**: classify cooling coil valve versus supply fan output

   In the cooling coil model, we assume we know the labels for all the points in each unit except those for "ChW VLV" and "SF SPEED". As is mentioned earlier, it is not commonplace for cooling coils in AHUs to have " CLC AIR InT, CLC AIR OutT". Hence, we will use "MAT" to approximate "CLC AIR InT", which can be a reasonable estimation when there is no heating coil for

the unit. We also use "SAT" to approximate "CLC AIR OutT" since they are close and there is not much heat exchange happening between the cooling coil air outlet and the supply air section, as is seen from the schematic diagram of the AHU in Figure 5.2. Notice that this is a decision we make due to the constraint of the data we have access to and, in practical use, we would suggest to instrument the necessary hardware if a very accurate model is desired.

Similar to the mixing box model, we calculate the predicted "CLC AIR OutT" following Equation 5.6 using "ChW VLV" and "SF SPEED", respectively. We are expecting that "ChW VLV" can generate a better prediction then "SF SPEED". Notice there are parameters we need to choose to use the model to make the predictions. Table 5.6 shows the values we choose for these parameters $\boldsymbol{\theta}$.

Parameter	**Value**
C_{pa}	1.005 kJ/(kg°C)
C_{pw}	4.1865 kJ/(kg°C)
Δt	15 min / 1 min
M_c	10 kg
C_c	0.91 kJ/(kg°C)
$\dot{m}_{w,c}^{max}$	0.38 kg/s
M_h	10 kg
C_h	0.91 kJ/(kg°C)
$\dot{m}_{w,h}^{max}$	0.38 kg/s

Table 5.6: The parameters for the cooling coil and heating coil model

Among all these parameters, C_{pa} and C_{pw} are specific heat for air and water with fixed values. The simulation time step Δt is set to be 15 minutes for the real-world data and 1 minute for the simulation data. For the parameters of cooling coils, the values should change for different units. However, as we do not have access to the design specification of each coil, we choose the values

empirically following the 4-row coil definition in [93] for all simulations. The same values are also used for the simulation of a heating coil. Utilizing data to estimate these parameters is left future work.

3. **Heating Coil**: classify heating coil valve versus supply fan output

   In the cooling coil model, we assume we know the labels for all the points in each unit except those for "HW VLV" and "SF SPEED". In the real-world dataset, we also do not have " HTC AIR InT, HTC AIR OutT". We can reasonably approximate " HTC AIR InT" using "MAT". However, the unit that has a heating coil also has a cooling coil, and there are no sensors instrumented to measure the air side temperature before the cooling coil after the heating coil. As a compromise, we assume the cooling coil has little impacts on the temperature change, and we use "SAT" to approximate "HTC AIR OutT".

   Similarly, we use the model to predict "HTC AIR OutT" following Equation 5.10 using the parameters in Table 5.6. In addition to recognizing the pair "HW VLV" versus "SF SPEED", we also try to recognize the pair "HW VLV" versus "ChW VLV" following the same logic.

5.5 Validation Results

5.5.1 Simulation Data

We are using the accuracy to measure the performance of the physical model-based approach to recognize point types. Table 5.7 shows the results for different scenarios with distinct fitness functions when using the simulation data. As we can see, the median absolute error and the mean absolute error perform better in all five tasks, being able to achieve an accuracy range from 78% to 100%.

	R^2	DTW	MAE	MSE	MedAE
MAT vs RAT	1.00	1.00	1.00	1.00	1.00
RAT vs SAT	0.72	0.50	1.00	0.72	0.83
ChW VLV vs SF SPEED	0.61	0.61	0.78	0.61	0.89
HW VLV vs SF SPEED	1.00	1.00	1.00	1.00	1.00
HW VLV vs ChW VLV	0.94	0.94	0.94	0.94	1.00

Table 5.7: Classification accuracy summary of different models using different metric functions with simulation data

To get a sense of how this approach differs from the existing approaches utilizing descriptive statistics, we calculate the mean and standard deviation of 18 mixed air temperature sensors and 18 return air temperature sensors and plot them in Figure 5.5 with different colors. Each point in the figure represents either "RAT" or "MAT". Using statistical quantities, two different types of sensors are separable in most cases. However, there are still points which are not distinguishable using statistical features, as is seen in the lower bottom section of the figure where two "MAT" sensors are treated as "RAT" sensors if these two statistical quantities are used. However, these points can be recognized correctly using the physical model-based approaches.

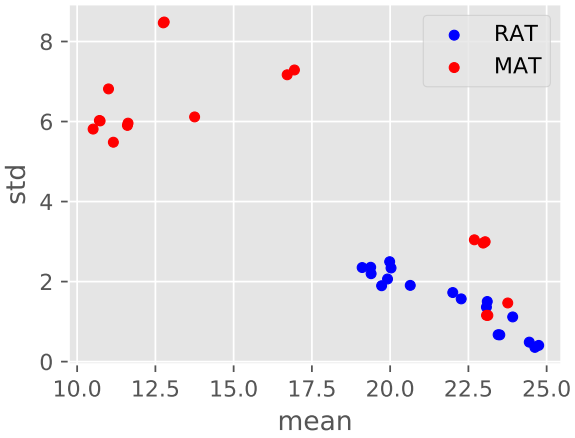


Figure 5.5: Scatter plot of mix air temperature sensors and return air temperature sensors in 2D using mean and standard deviation

To understand the predictive power of the model, Figure 5.6 shows the time series plots of the predicted value versus the actual values for the cooling coil outlet air temperature. The predicted values are almost the same as compared to the actual values, indicating the model can generate one variable based on other variables. The predictive power of the model-based on other associated variables is the essential reason that helps us to produce the label assignments correctly for the points.

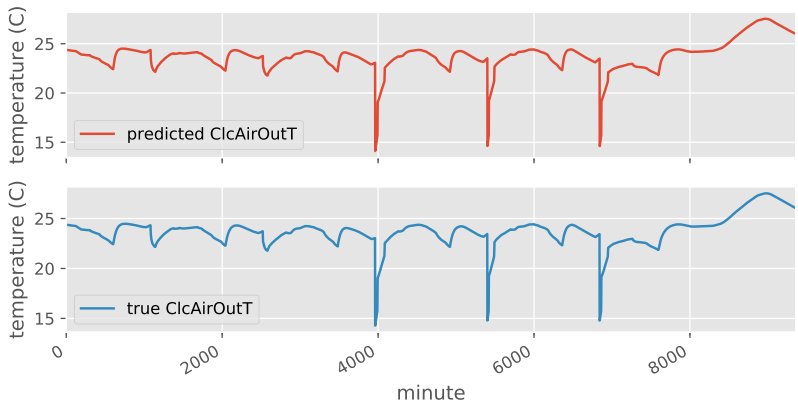


Figure 5.6: Time series plots of predicted values and actual values using simulation data

We now examine how each model performs using the real-world data in the next subsection.

5.5.2 Real-world Data

Table 5.8 shows the accuracy for different models using each of the fitness functions for the real-world dataset. As we can see, overall, the highest accuracy can be achieved when we use R^2 as the fitness function to discriminate different point types. It has yielded the highest accuracy for three out of five models. The accuracy 87% in the first row suggests that we can recognize the types being "MAT" or "RAT" correctly from time series data 87% of the time among 384 testing instances. It is worth pointing out this model does not rely on the training data of other "MAT" or "RAT". Instead, we are using the information from other BAS points related to

these sensors to identify them. The identity is defined based on the relationship of this BAS point with other BAS points instead of solely depending on the patterns of the time series data.

	$\mathbf{R}^2$	**DTW**	**MAE**	**MSE**	**MedAE**
MAT vs RAT	0.87	0.59	0.64	0.56	0.77
RAT vs SAT	0.65	0.63	0.69	0.65	0.73
ChW VLV vs SF SPEED	0.75	0.66	0.81	0.75	0.83
HW VLV vs SF SPEED	0.83	0.67	0.83	0.83	0.75
HW VLV vs ChW VLV	0.83	0.75	0.83	0.83	0.75

Table 5.8: Classification accuracy summary of different models using different metric functions with real-world data

We also observe that MedAE performs better in recognizing "RAT versus SAT" and "ChW VLV versus SF SPEED". One explanation is that the median absolute error only considers the median error loss, which is more robust to outliers and the data with extreme values. This could be the case for classifying "RAT versus SAT" and "ChW VLV versus SF SPEED" where the R^2 score is not performing the best.

Mixing Box

We further examine what are the values of metrics when different label assignments are made for the mixing box from one unit. Figure 5.7 shows the scatter plots of the true values versus the predicted values from three label assignments including when we assign the label correctly (base), when we confuse "MAT" with "RAT" (A1), and when we confuse "RAT" with "SAT" (A2). Since the base one is generated from the true label assignments, when the highest R^2 score is associated with it, this indicates we have produced a correct label assignment, i.e., recognizing the point types correctly from the confused pairs. The negative R^2 scores indicate the model fits the data poorly and produces a larger sum of squares of residuals than the total sum of squares. This can happen when we force the data to follow a

certain linear model where we do not have an intercept term, which is the case in our mixing box model. It could also happen when the model is nonlinear, which we will see in the coil models.

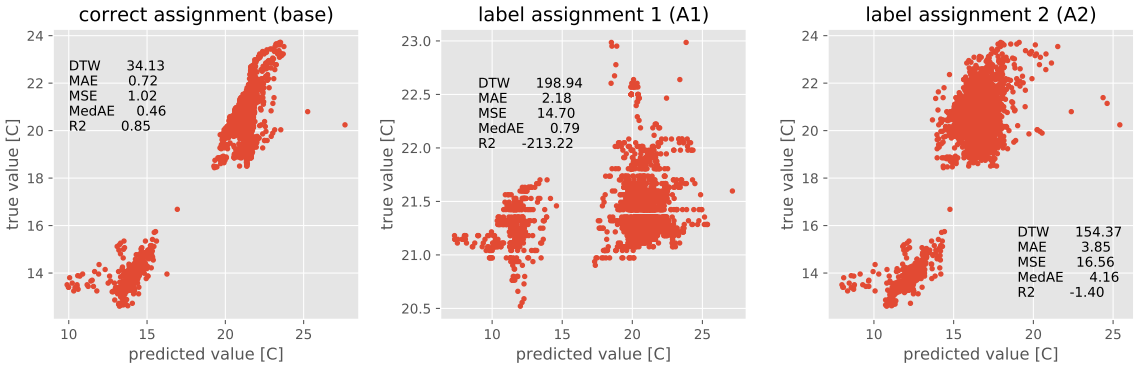


Figure 5.7: Scatter plots for three label assignments of mixing box where the highest R^2 score represents the most probable assignment. The base one is the correct assignments; A1 and A2 are generated based on two incorrect label assignments

Figure 5.8 shows the scatter plots from three label assignments for another unit. We encounter negative R^2 scores for all three assignments and the one with the highest score is not the base one, indicating the model makes an incorrect assignment. To understand why the model makes an incorrect assignment, we plot the raw time series for the sensors in this unit in Figure 5.9.

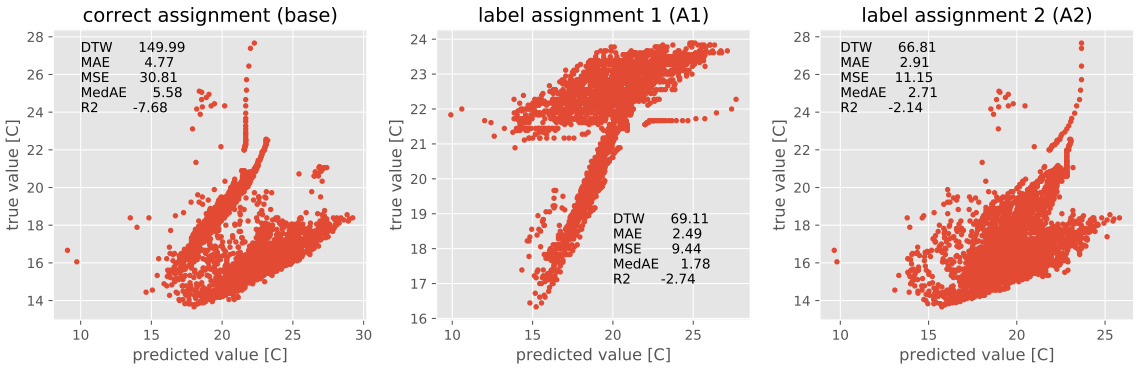


Figure 5.8: Scatter plots for three label assignments of mixing box with incorrect inference

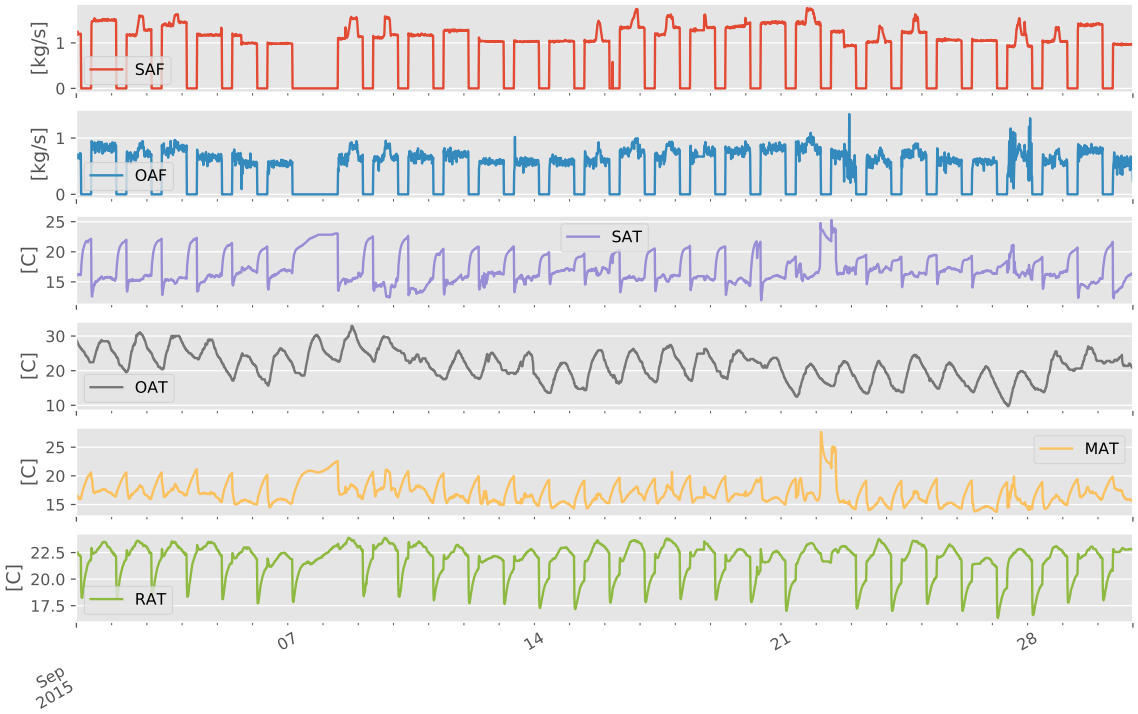


Figure 5.9: Raw time series plots of points in the mixing box with incorrect inference

As we can see from the raw time series, "SAF" and "OAF" turn into 0 during night hours. This can happen when the outdoor air damper is completely closed, and there is no air flowing in the AHU duct. This is not common in our dataset as most HVAC systems maintain a minimum amount of air flowing even during the night time when the HVAC system is shut down, e.g., there is no fan running, and the cooling/heating valves are closed while the natural ventilation is on. When the flow rates are 0 during the night, our physical model does not hold anymore. Let us revisit the model for the mixed air temperature:

$$T_{mix} = \frac{\dot{m}_{out}}{\dot{m}_{sup}} T_{out} + (1 - \frac{\dot{m}_{out}}{\dot{m}_{sup}}) T_{ret} \tag{5.21}$$

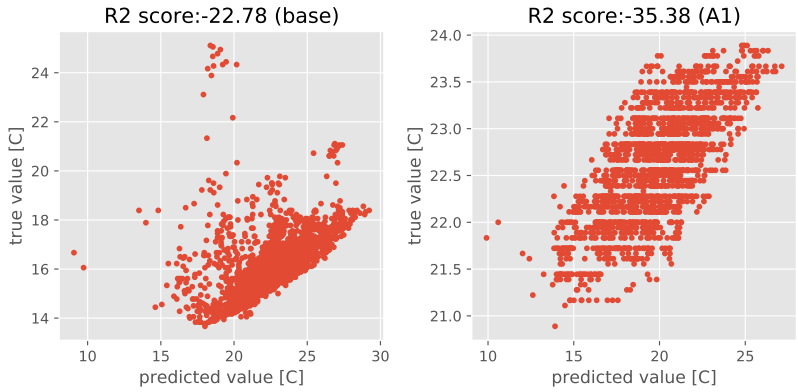


Figure 5.10: Scatter plots for the model after removing 0 values from the flow rate measurements

When "SAF" and "OAF" are 0, the ratio $\frac{\dot{m}_{out}}{\dot{m}_{sup}}$ is not defined, which leads to the abnormal behaviors of the model. In our implementation, we let $T_{mix} = T_{ret}$ as long as $\dot{m}_{sup} = 0$, and $T_{mix} = T_{out}$ if $\dot{m}_{ret} = 0$ and $\dot{m}_{sup} \neq 0$. However, such an approximation might still not reflect the actual behavior. Hence we further ignore all the time steps where the flow rates are 0 and only use the data during the daytime when there is air flowing. The resulting data is shown in scatter plots in Figure 5.10. As we can see the R^2 score is still negative, but the base assignment has a higher score than A1 assignment.

Cooling Coil

For the cooling coil model, we only consider two assignments where the base represents the true label assignment, and A1 represents the case when we confuse "ChW VLV" with "SF SPEED". Figure 5.11 shows the scatter plots of these two scenarios. As we can see, the predicted values based on "ChW VLV" fit the model better than that of "SF SPEED", which is justified by each one of the five metrics.

Figure 5.12 also shows four scatter plots when incorrect assignments are produced. Although the difference between R^2 scores is subtle, the two examples shown

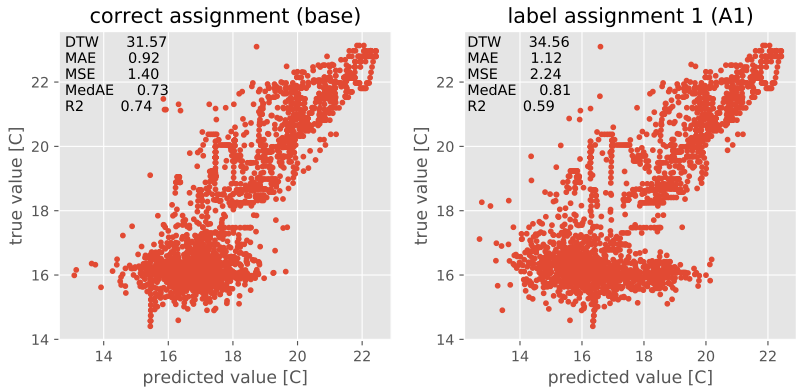


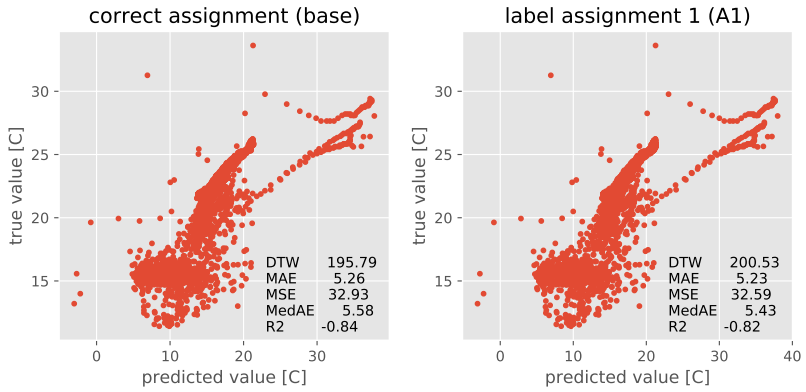
Figure 5.11: Scatter plots for two label assignments of cooling coil where the highest R^2 score represents the most probable assignment. The base one is the correct assignments; A1 is generated based on incorrect label assignments.

in the figures suggest that the physical model can sometimes fail. To understand why in Figure 5.13 we also plot the raw time series for one unit where the incorrect assignments are made.

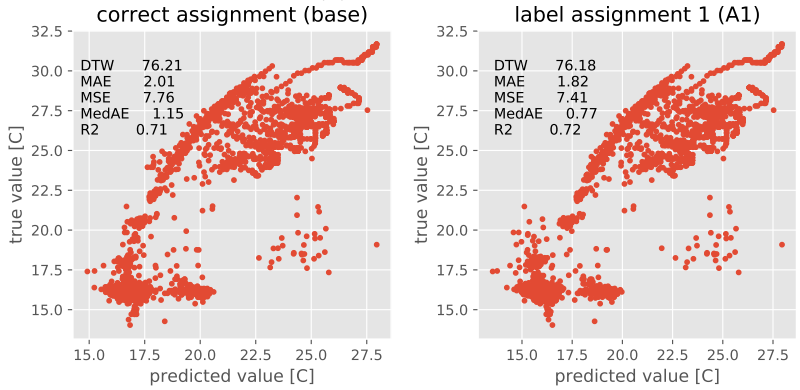
As we can see "ChW VLV" and "SF SPEED" indeed have similar patterns and ranges based on time series data, which leads to the similar predicted values for "CLC AIR OutT" and further similar R^2 scores. To understand why the similar predicted values are generated, notice "CLC AIR OutT" or T_c for the next timetick is calculated based on $\frac{dT_c}{dt}$ and T_c at the current timetick. When "ChW VLV" and "SF SPEED' are used to predict "CLC AIR OutT" respectively, the same initial value $T_c^{t_0}$ is provided and the variable that affects the T_c is $\frac{dT_c}{dt}$ defined as:

$$\frac{dT_c}{dt} = -\frac{\dot{m}_{sup}C_{pa}}{M_cC_c}T_c + \frac{\dot{m}_{sup}C_{pa}T_{ai,c} - \dot{m}_{w,c}^{max}{u_c}^2 \cdot C_{pw}(T_{wo,c} - T_{wi,c})}{M_cC_c} \tag{5.22}$$

In the base case, the correct values of "ChW VLV" are fed into u_c to calculate $\frac{dT_c}{dt}$ (base); and in the case of A1, the values from "SF SPEED" are fed into u_c to calculate $\frac{dT_c}{dt}$ (A1). As the predicted values of T_c is affected by $\frac{dT_c}{dt}$, we plot how



(a) Example 1



(b) Example 2

Figure 5.12: Scatter plots of two examples for points in cooling coil with incorrect inference

this quantity changes over time for two label assignments in Figure 5.14a. It is clear that both assignment have very similar gradient values over time. If we further plot $\frac{dT_c}{dt}$ (base) against $\frac{dT_c}{dt}$ (A1) in Figure 5.14b, we can clearly see a straight line. Such a high similarity of $\frac{dT_c}{dt}$ for two label assignments is the main reason that the physical model fails to recognize points of different types.

Heating Coil

Figure 5.15 shows the scatter plots of the true values versus the predicted values from three label assignments including when we assign the label correctly (base) when we confuse "HW VLV" as "SF SPEED" (A1), and when we confuse "HW VLV" as "ChW VLV" (A2). The base one having the highest R^2 is the correct

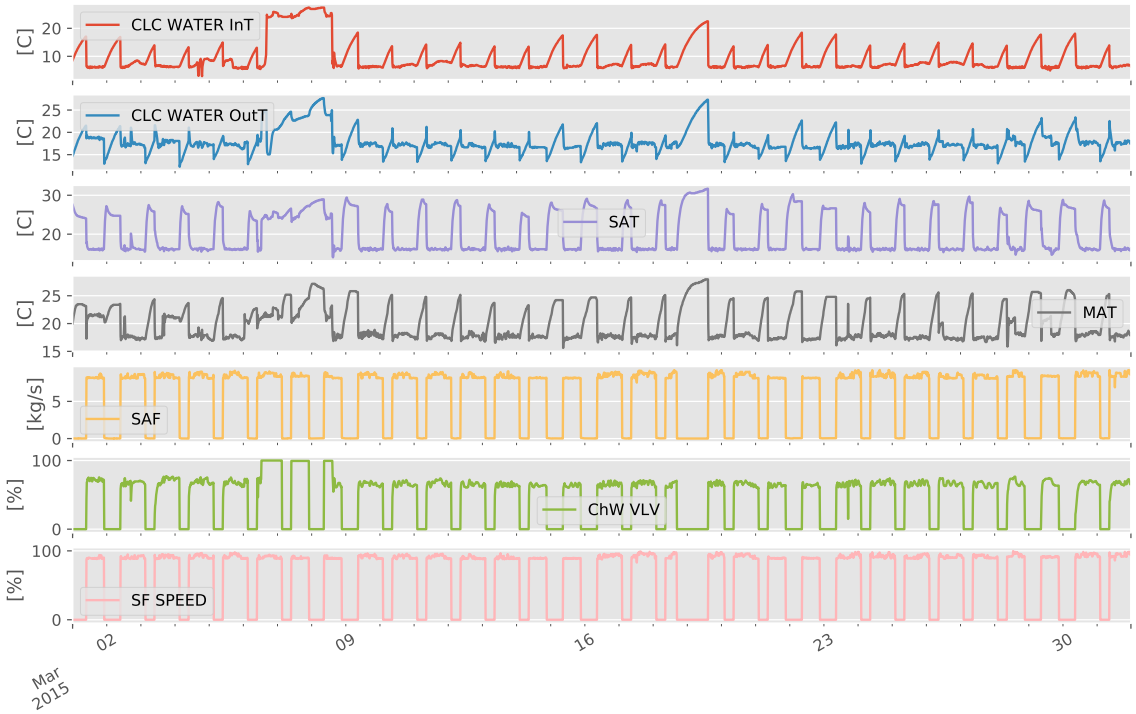
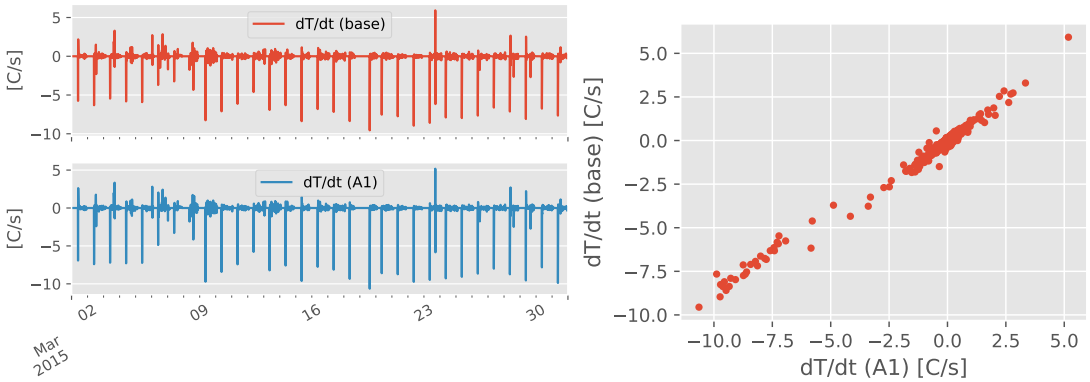


Figure 5.13: Raw time series plots of points in the cooling coil with incorrect inference



(a) Time series plots of $\frac{dT}{dt}$ (b) Scatter plots of $\frac{dT}{dt}$

Figure 5.14: Time series and scatter plots of $\frac{dT}{dt}$ for two label assignments

assignment.

Figure 5.16 shows the scatter plots for another unit when the incorrect assignments are made. These three scatter plots do also have similar patterns. Hence we plot the raw time series of eight BAS points for this unit in Figure 5.17. As is seen

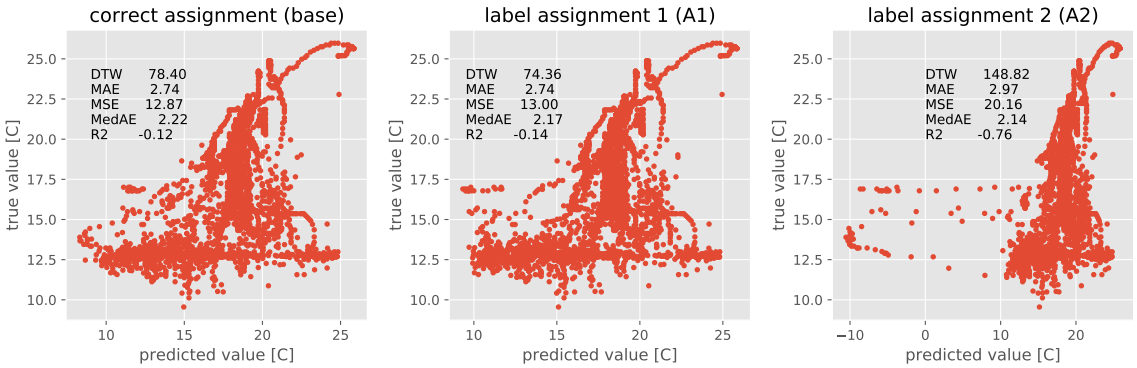


Figure 5.15: Scatter plots for three label assignments of heating coil where the highest R^2 score represents the most probable assignment. The base one is the correct assignments; A1 and A2 are generated based on incorrect label assignments

in the figure, "HW VLV", "SF SPEED" and "ChW VLV" do have similar patterns and range, which make them indistinguishable sometimes. We could also reason similarly by computing $\frac{dT_h}{dt}$ for three different labels assignments.

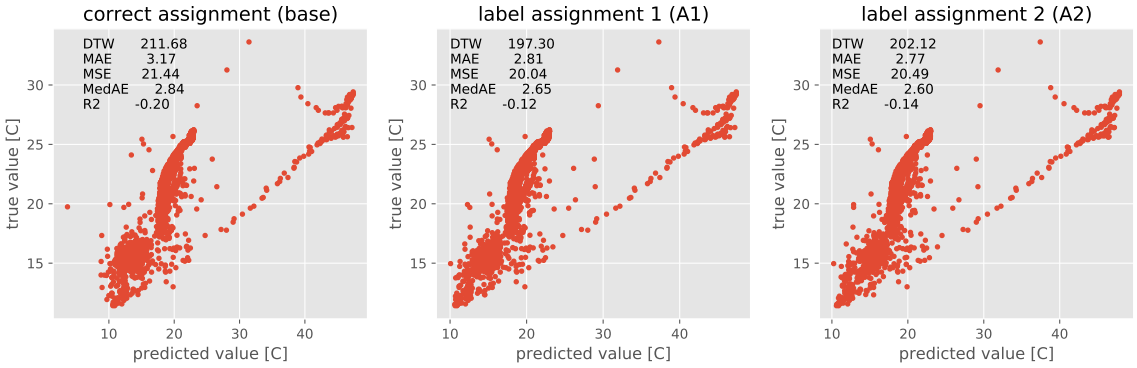


Figure 5.16: Scatter plots for three label assignments of heating coil with incorrect inference

Meanwhile, we notice even in the case of a correct assignment of the model, a negative R^2 score is generated. The negative R^2 score is because the model being used is based on a non-linear differential equation, which could generate predictions worse than a straight line of the average of the true values. Another reason is that we use 'SAT" to approximate "HTC AIR OutT" which might not be valid in many cases. For example, we assume this could be true during the winter when the chilled

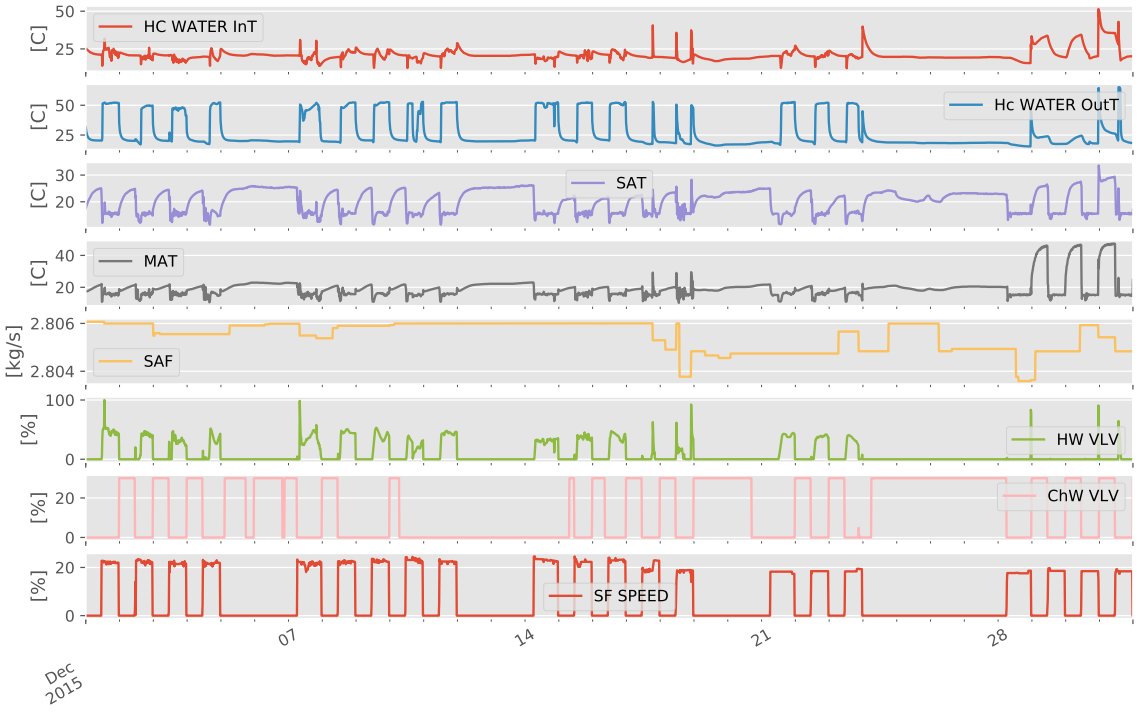


Figure 5.17: Raw time series plots of points in the heating coil with incorrect inference

water valve is fully closed. However, as is seen in Figure 5.17, the chilled water valve can still be open even during the winter. Additionally, we only have one unit from the real-world dataset. The observation and conclusion from this specific heating coil might not also hold for other heating coils.

5.6 Discussions and Limitations

The physical model-based approach has been demonstrated to be feasible to infer the type of BAS points in AHUs. One advantage is that we do not require training data the same way as data-driven models. In other words, the identity of certain time series is determined by the relationship with other time series in the same unit, instead of being determined by other times series of the same type from different units. This poses some advantages as the time series of the same type from different units might have very different patterns, but the relationships of points in

the same units will be largely dictated by physics. This can potentially complement the data-driven models where we first infer the BAS points which we are certainly based on the probability interpretation, and for the points we are uncertain about, we can utilize physical models to discriminate them. Furthermore, when the model fails, we can reason based on physical principles, e.g., whether the assumptions of the model holds, whether the model can describe the actual system, whether certain time series are distinguishable using time series data alone, etc.

However, this approach does have limitations such as the following:

- It needs prior knowledge of which sensors are in the same unit. This will not be a problem when such information can be derived based on tags. For example, as is seen earlier in Table 1.3, the points which are in the same unit may have the same prefix. However, for some other units with uncommon naming conventions for tags, such information might not be able to be extracted. The metadata inference approaches to derive the equipment and location information [51, 52, 53, 44, 54] could potentially be used.

- It needs to assume a portion of BAS points has been labeled to infer the remaining unknown points, which helps reduce the possible permutations of label assignments. The need of labeled points is from the practical implementation perspectives as it reduces the complexity of the approach and improves the confidence level of the results.

- It needs to make sure the BAS has all the variables required by the models. Otherwise, we either disregard the model or approximate the required variables in the model. The approximation could impact the performance of the approach, which can be seen in the heating coil and cooling coil examples.

- It needs to use the additional parameters, for example, the design maximum flow rate, the size and the dimension of the cooling/heating coil. It takes the effort to collect these data and sometimes they are even unavailable. When we make some assumptions and set the value heuristically, the model might not reflect the actual system behavior.

5.7 Conclusion

In this chapter, we extend the existing work of metadata inference approaches by introducing a new approach based on physical models. The approach has been tested and validated on both the simulation data and the real-world data to infer the types of BAS points inside the mixing box, the cooling coil and the heating coil of an AHU. The resulting accuracy ranges from 73% to 100% regarding classifying the easily confused types in data-driven approaches. Using the physical models, we can understand how the data values of BAS points are generated and how the relationship among points can help the identification task. We can further diagnose the model when it makes incorrect label assignments.

The physical model-based approach has its limitations as it relies on the information from BAS points that are known to be functionally tied together by a specific building system. To address this limitation, a promising future direction to work on is to integrate the physical model-based approach with the data-driven approaches, which may generate more metadata information to facilitate the deployment of FDD applications. Additionally, some other future directions could be studied including making use of virtual sensing technology to approximate some values which are not available in BAS but are required by the models, and development of grey-box models to estimate the model parameters from the data.

Chapter 6

Conclusions

In this dissertation, we focus on the use of time series data obtained from sensors and actuators in buildings to infer the associated metadata information. In particular, we develop a metadata inference framework to provide operational information support such that the manual efforts to acquire this required information can be reduced by computerized algorithms based on metadata inference approaches. This will decrease the cost of deploying FDD applications on multiple buildings and further bring more benefits to building managers. Moreover, it is important to note that metadata inference is not only useful as a one-time effort, since it can also be used to verify and re-tune metadata throughout the life of the building and may even be used for security purposes (i.e., to ensure that the reported time series values are behaving as expected) if one is concerned with unauthorized tampering with the BAS.

The main conclusions of this thesis are briefly outlined below.

1. **Understanding the required BAS points and associated metadata for FDD approaches in secondary HVAC systems guides the metadata**

inference task.

(a) The most commonly required BAS points by FDD algorithms are identified, of which six points in AHUs are used by more than 30% of FDD approaches including sensors monitoring supply air temperature, outside air temperature, chilled water valve position, return air temperature, supply air flow rate, and mixed air temperature. These identified BAS points provide guidance regarding what metadata should be inferred using the metadata inference approaches.

(b) Data-driven models are more prevalent which occupies 62% of total approaches reviewed (68 out of 110), and 82% of developed FDD approaches (90 out of 110) can be applied to AHUs.

(c) The overall distributions of frequent point types existing in BASs and required by FDD approaches are similar where they share the same 12 out of 20 types. The identified BAS points for different FDD approaches can help building managers to select which approaches are applicable to the buildings being managed. It also provides guidance regarding what hardware should be instrumented if FDD applications are desired for a specific building.

2. **The existing metadata inference approaches can be generalized to multiple building sites.**

(a) The average performance of these approaches in terms of accuracy is similar across building sites, though there is a variance for different building sites given the difference in the points distribution.

(b) The expected accuracy of classifying the type of points required by a particular FDD application (APAR) for a new unseen building is, on average, 75%.

(c) The performance of the metadata inference approach does not decrease as long as training data and testing data are extracted from adjacent months.

(d) The coverage and tolerance accuracy based on probabilistic interpretations can provide useful information to building operators and managers who need to label BAS points in buildings, as they can trust the predictions with high probabilities and reduce the searching scope to focus only on the points which have uncertain predictions from the model.

3. **A physical model-based metadata inference approach can complement existing data-driven models and provide physical interpretations in the case of incorrect metadata being inferred.**

(a) The approach can demonstrate how the data values of BAS points are generated and how the relationship among BAS points in the same unit can affect each other based on physical principles.

(b) The approach has shown its capabilities to discriminate the BAS points that are easily confused in data-driven based metadata inference approaches.

(c) Diagnosis can be conducted on the approach when an incorrect prediction of the metadata is produced.

(d) The approach is developed based on the physical principles which do not require the training data.

Despite all these positive findings, this line of research work is still far from fully minimizing the cost of implementation for FDD applications. The work presented in this dissertation serves as a starting point for different new avenues of research, including:

1. There is a need to evaluate tag-based approaches, as well as active approaches to infer the required metadata in large scale in addition to time series based approaches.

2. A study to quantify the economic gain and benefits of using metadata inference approaches when implementing FDD applications in real buildings is needed.

3. Simulation models for buildings could be used to generate datasets in a controlled environment for the study of metadata inference approaches. This clean dataset can be further corrupted to resemble real-world data, but allowing one to control for many of the uncontrollable variables such as the presence of faults in the training datasets.

4. The physical model-based approach could be combined with the data-driven models to build a grey-box model to improve the capabilities and limitations of existing approaches.

5. A user-friendly interface could be developed to bridge the metadata being inferred and the implementation of FDD applications.

References

[1] E. I. Administration, "Annual energy outlook 2015," 2015, https://www.eia.gov/outlooks/archive/aeo15/.

[2] A. Dexter, J. Pakanen *et al.*, "Demonstrating automated fault detection and diagnosis methods in real buildings," 2001.

[3] K. W. Roth, D. Westphalen, M. Y. Feng, P. Llana, and L. Quartararo, "Energy impact of commercial building controls and performance diagnostics: market characterization, energy impact of building faults and energy savings potential," *TIAX Report D*, vol. 180, 2005.

[4] M. R. Brambley, P. Haves, S. C. McDonald, P. A. Torcellini, D. Hansen, D. Holmberg, and K. Roth, *Advanced sensors and controls for building applications: Market assessment and potential R & D pathways.* Pacific Northwest National Laboratory Washington, DC, USA, 2005.

[5] R. Jagpal, "Computer aided evaluation of HVAC system performance: Technical synthesis report," International Energy Agency, Tech. Rep., 2006.

[6] K. Roth, D. Westphalen, P. Llana, and M. Feng, "The Energy Impact of Faults in US Commercial Buildings," *International Refrigeration and*

Air Conditioning Conference, pp. 600–609, 2004. [Online]. Available: http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1664&context=iracc

[7] C. I. Agency, "The world factbook, 2017," 2017, https://www.cia.gov/library/publications/the-world-factbook/index.html.

[8] M. Liddament, "Technical synthesis report: Real time simulation of hvac systems for building optimisation, fault detection and diagnostics," *Coventry, UK, ESSU*, 1999.

[9] S. Katipamula and M. R. Brambley, "Review article: Methods for fault detection, diagnostics, and prognostics for building systems-a review, part i," *HVAC&R Research*, vol. 11, no. 1, pp. 3–25, 2005. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10789669.2005.10391123

[10] ——, "Review article: Methods for fault detection, diagnostics, and prognostics for building systems-ÂĂÂŤa review, part ii," *HVAC&R Research*, vol. 11, no. 2, pp. 169–187, 2005. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10789669.2005.10391133

[11] K. Bruton, P. Raftery, and B. Kennedy, "Review of automated fault detection and diagnostic tools in air handling units," *Energy Efficiency*, 2014. [Online]. Available: http://link.springer.com/article/10.1007/s12053-013-9238-2

[12] N. E. Klepeis, W. C. Nelson, W. R. Ott, J. P. Robinson, A. M. Tsang, P. Switzer, J. V. Behar, S. C. Hern, and W. H. Engelmann, "The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants." *Journal of exposure analysis and environmental*

epidemiology, vol. 11, no. 3, pp. 231–52, jan 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/11477521

[13] S.-h. Lee, "Barriers to application of fault detection and diagnosis (fdd) techniques to air-conditioning systems in buildings in hong kong," Ph.D. dissertation, The Hong Kong Polytechnic University, 2010.

[14] J. M. House, H. Vaezi-Nejad, and J. M. Whitcomb, "An expert rule set for fault detection in air-handling units," *Transactions-American Society of Heating Refrigerating and Air Conditioning Engineers*, vol. 107, pp. 858–874, 2001.

[15] J. Schein, S. T. Bushby, N. S. Castro, and J. M. House, "A rule-based fault detection method for air handling units," *Energy and Buildings*, vol. 38, no. 12, pp. 1485–1492, dec 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778806001034

[16] A. Ahmed, J. Ploennigs, K. Menzel, and B. Cahill, "Multi-dimensional building performance data management for continuous commissioning," *Advanced Engineering Informatics*, vol. 24, no. 4, pp. 466–475, 2010.

[17] S. Kaldorf and P. Gruber, "Practical experiences from developing and implementing an expert system diagnostic tool/discussion," *ASHRAE Transactions*, vol. 108, p. 826, 2002.

[18] W. Livingood, J. Stein, T. Considine, and C. Sloup, "Review of current data exchange practices: Providing descriptive data to assist with building operations decisions," *Contract*, vol. 303, pp. 275–3000, 2011.

[19] Y. Park, "Point Naming Standards: A Necessary Evil for Building Information Integration," in *ISA Automation Week 2012: Control Performance*, 2012.

[20] DOE, "Building Energy Data Exchange Specification Scoping Report," 2013. [Online]. Available: http://energy.gov/eere/buildings/downloads/building-energy-data-exchange-specification-scoping-report

[21] L. Luskay, M. Brambley, and S. Katipamula, "Methods for Automated and Continuous Commissioning of Building Systems," Oak Ridge Operations, Oak Ridge, TN, Tech. Rep., Apr. 2003. [Online]. Available: http://www.osti.gov/scitech/biblio/810800

[22] J. F. Butler and R. Veelenturf, "Point naming standards," *ASHRAE Journal*, vol. 52, p. B16, 2010.

[23] A. Chen and C. Talon, "Data Integration for Intelligent Buildings," 2016. [Online]. Available: https://www.navigantresearch.com/research/data-integration-for-intelligent-buildings

[24] A. Schumann, J. Ploennigs, and B. Gorman, "Towards automating the deployment of energy saving approaches in buildings," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings - BuildSys '14*. New York, New York, USA: ACM Press, nov 2014, pp. 164–167.

[25] J. Ploennigs, B. Hensel, H. Dibowski, and K. Kabitzsch, "Basont - a modular, adaptive building automation system ontology," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct 2012, pp. 4827–4833.

[26] Haystack, "Project haystack," http://project-haystack.org/, 2014, accessed: 2015-07-23.

[27] A. Bhattacharya, J. Ploennigs, and D. Culler, "Short Paper: Analyzing Metadata Schemas for Buildings," in *Proceedings of the 2nd ACM International*

Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15. New York, New York, USA: ACM Press, nov 2015, pp. 33–34. [Online]. Available: http://dl.acm.org/citation.cfm?id=2821650.2821669

[28] X. Liu, B. Akinci, J. H. Garrett, Jr, and M. Bergés, "Requirements for an integrated framework of self-managing hvac systems," in *Computing in Civil Engineering*, 2011, pp. 802–809.

[29] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R. Gupta, M. B. Kjærgaard, M. Srivastava, and K. Whitehouse, "Brick: Towards a unified metadata schema for buildings," in *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, ser. BuildSys '16. New York, NY, USA: ACM, 2016, pp. 41–50.

[30] F. Leonardi, H. M. Reeve, T. C. Wagner, Z. Xiong, and W. June, "Assisted Point Mapping to Enable Cost-effective Deployment of Intelligent Building Applications," in *International Compressor Engineering, Refrigeration and Air Conditioning, and High Performance Buildings Conferences*, 2016, pp. 1–8.

[31] V. Bazjanac and D. Crawley, "Industry foundation classes and interoperable commercial software in support of design of energy-efficient buildings," *Proceedings of Building Simulation'99*, 1999. [Online]. Available: http://www.inive.org/members{_}area/medias/pdf/Inive/IBPSA/UFSC755.pdf

[32] M. Botts and A. Robin, "OpenGIS sensor model language (SensorML) implementation specification," *OpenGIS Implementation Specification OGC*, vol. 7, no. 000, 2007.

[33] N. Dawes, K. A. Kumar, S. Michel, K. Aberer, and M. Lehning, "Sensor Metadata Management and Its Application in Collaborative Environmental Research," in *2008 IEEE Fourth International Conference on eScience*. IEEE, dec 2008, pp. 143–150.

[34] J. F. Butler, "Point naming standards," *ASHRAE Journal*, vol. 52, p. B16, 2010.

[35] S. Roth, "Open green building xml schema: A building information modeling solution for our green world, gbxml schema," 2014.

[36] V. Charpenay, S. Kabisch, D. Anicic, and H. Kosch, "An ontology design pattern for iot device tagging systems," in *2015 5th International Conference on the Internet of Things (IOT)*, Oct 2015, pp. 138–145.

[37] X. Liu and B. Akinci, "Requirements and Evaluation of Standards for Integration of Sensor Data with Building Information Models," in *Computing in Civil Engineering (2009)*. ASCE, 2009, pp. 95–104.

[38] A. Bhattacharya, J. Ploennigs, and D. Culler, "Short Paper: Analyzing Metadata Schemas for Buildings," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*. New York, New York, USA: ACM Press, nov 2015, pp. 33–34.

[39] E. Holmegaard, A. Johansen, and M. B. Kjargaard, "Towards a metadata discovery, maintenance and validation process to support applications that improve the energy performance of buildings," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, mar 2016, pp. 1–6.

[40] B. Balaji, C. Verma, B. Narayanaswamy, and Y. Agarwal, "Zodiac: Organizing Large Deployment of Sensors to Create Reusable Applications for Buildings," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*. New York, New York, USA: ACM Press, nov 2015, pp. 13–22.

[41] D. Hong, H. Wang, J. Ortiz, and K. Whitehouse, "The Building Adapter," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*. New York, New York, USA: ACM Press, nov 2015, pp. 123–132.

[42] J. Gao, J. Ploennigs, and M. Berg'es, "A data-driven meta-data inference framework for building automation systems," in *Proceedings of the 2nd ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ser. BuildSys '15. New York, NY, USA: ACM, 2015. [Online]. Available: http://dx.doi.org/10.1145/2821650.2821670

[43] J. P. Calbimonte, Z. Yan, H. Jeung, O. Corcho, and K. Aberer, "Deriving semantic sensor metadata from raw measurements," in *Proceedings of the 5th International Conference on Semantic Sensor Networks - Volume 904*, ser. SSN'12. Aachen, Germany, Germany: CEUR-WS.org, 2012, pp. 33–48.

[44] M. Koc, B. Akinci, and M. Bergés, "Comparison of linear correlation and a statistical dependency measure for inferring spatial relation of temperature sensors in buildings," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings - BuildSys '14*. New York, New York, USA: ACM Press, nov 2014, pp. 152–155.

[45] E. Holmegaard and M. B. Kjærgaard, "Mining Building Metadata by Data Stream Comparison," in *Proceeding of the 2016 IEEE Conference on Technologies for Sustainability*, 2016, pp. 28–33.

[46] D. Hong, Q. Gu, and K. Whitehouse, "High-dimensional Time Series Clustering via Cross-Predictability," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, 2017, pp. 642–651. [Online]. Available: http://proceedings.mlr.press/v54/hong17a.html

[47] A. A. Bhattacharya, D. Hong, D. Culler, J. Ortiz, K. Whitehouse, and E. Wu, "Automated Metadata Construction to Support Portable Building Applications," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*. New York, New York, USA: ACM Press, nov 2015, pp. 3–12.

[48] M. Pritoni, A. A. Bhattacharya, D. Culler, and M. Modera, "Short Paper: A Method for Discovering Functional Relationships Between Air Handling Units and Variable-Air-Volume Boxes From Sensor Data," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*. New York, New York, USA: ACM Press, 2015, pp. 133–136.

[49] J. Koh, B. Balaji, V. Akhlaghi, Y. Agarwal, and R. Gupta, "Quiver: Using control perturbations to increase the observability of sensor data in smart buildings," *CoRR*, vol. abs/1601.07260, 2016. [Online]. Available: http://arxiv.org/abs/1601.07260

[50] D. Hong, H. Wang, K. Whitehouse, and S. Art, "Clustering-based Active Learning on Sensor Type Classification in Buildings," in *The 24th ACM International Conference on Information and Knowledge Management*. New York, New York, USA: ACM Press, 2015, pp. 363–372.

[51] C. Ellis, J. Scott, I. Constandache, and M. Hazas, "Creating a room connectivity graph of a building from per-room sensor units," in *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings - BuildSys '12*. New York, New York, USA: ACM Press, nov 2012, p. 177.

[52] J. Lu and K. Whitehouse, "Smart blueprints: automatically generated maps of homes and the devices within them," in *International Conference on Pervasive Computing*. Springer, 2012, pp. 125–142.

[53] D. Hong, J. Ortiz, K. Whitehouse, and D. Culler, "Towards automatic spatial verification of sensor placement in buildings," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. ACM, 2013, pp. 1–8.

[54] B. Akinci, M. Berges, and A. G. Rivera, "Exploratory Study Towards Streamlining the Identification of Sensor Locations Within a Facility," in *Computing in Civil and Building Engineering (2014)*. ASCE, 2014, pp. 1820–1827.

[55] F. Xiao and C. Fan, "Data mining in building automation system for improving building operational performance," *Energy and buildings*, vol. 75, pp. 109–118, 2014.

[56] R. Fontugne, J. Ortiz, D. Culler, and H. Esaki, "Empirical mode decomposition for intrinsic-relationship extraction in large sensor deployments," in *Workshop on Internet of Things Applications, IoT-App*, vol. 12, 2012.

[57] X. Liu, "An integrated information support framework for performance analysis and improvement of secondary hvac systems," Ph.D. dissertation, 2012, copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2016-03-11. [Online]. Available: https://search.proquest.com/docview/1221544166?accountid=9902

[58] Y. Yu, D. Woradechjumroen, and D. Yu, "A review of fault detection and diagnosis methodologies on air-handling units," *Energy and Buildings*, vol. 82, pp. 550–562, 2014.

[59] M. Padilla, "A review of fault detection, diagnosis and isolation methods for vav-air handling units," 2014.

[60] G. S. Okochi and Y. Yao, "A review of recent developments and technological advancements of variable-air-volume (VAV) air-conditioning systems," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 784–817, 2016.

[61] W. Kim and S. Katipamula, "A review of fault detection and diagnostics methods for building systems," *Science and Technology for the Built Environment*, pp. 1–18, apr 2017. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/23744731.2017.1318008

[62] M. Corsi and D. Choiniere, "A BEMS-assisted commissioning tool to improve the energy performance of HVAC systems," 2003. [Online]. Available: http://oaktrust.library.tamu.edu/handle/1969.1/5201

[63] D. Choinière, "DABO: A BEMS assisted ongoing commissioning tool." *National Conference on Building Commissioning: April*, 2008. [Online]. Available: http://www.bcxa.org/ncbc/2008/docs/Choiniere.pdf

[64] J. Schein and S. Bushby, "A Hierarchical Rule-Based Fault Detection and Diagnostic Method for HVAC Systems," *HVAC&R Research*, vol. 12, no. December 2014, pp. 111–125, 2006.

[65] S. Wang and F. Xiao, "AHU sensor fault diagnosis using principal component analysis method," *Energy and Buildings*, vol. 36, no. 2, pp. 147–160, 2004.

[66] J. Wix and J. Karlshoej, "Information delivery manual: Guide to components and development methods," *BuildingSMART International*, 2010.

[67] H. Beyer and K. Holtzblatt, *Contextual design: defining customer-centered systems*. Elsevier, 1997.

[68] H. Dibowski, J. Vass, O. Holub, and J. Rojicek, "Automatic setup of fault detection algorithms in building and home automation," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, sep 2016, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7733622/

[69] F. Xiao, S. Wang, X. Xu, and G. Ge, "An isolation enhanced PCA method with expert-based multivariate decoupling for sensor FDD in air-conditioning systems," *Applied Thermal Engineering*, vol. 29, no. 4, pp. 712–722,

mar 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1359431108001774

[70] X. Yi, Y. Chen, and L. Wu, "Sensor fault detection and diagnosis for VAV system based on principal component analysis," *Proceedings: building simulation*, 2007. [Online]. Available: http://www.ibpsa.org/proceedings/BS2007/p132{_}final.pdf

[71] S. Li and J. Wen, "A model-based fault detection and diagnostic methodology based on PCA method and wavelet transform," *Energy and Buildings*, vol. 68, no. PARTA, pp. 63–71, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.enbuild.2013.08.044

[72] H. Yoshida, T. Iwami, H. Yuzawa, and M. Suzuki, "Typical faults of air conditioning systems and fault detection by ARX model and extended Kalman filter," 1996. [Online]. Available: https://www.osti.gov/scitech/biblio/392482

[73] J. Gao and M. Berges, "A large-scale evaluation of automated metadata inference approaches on sensors from hundreds of air handling units," *Submitted to Advanced Engineering Informatics*, 2017.

[74] W. S. Li and C. Clifton, "Semantic integration in heterogeneous databases using neural networks," in *VLDB*, vol. 94, 1994, pp. 12–15.

[75] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal*, vol. 10, no. 4, pp. 334–350, dec 2001.

[76] N. Japkowicz, *Assessment Metrics for Imbalanced Learning*. John Wiley & Sons, Inc., 2013, pp. 187–206.

[77] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, p. 49, nov 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1882471.1882479

[78] D. J. Hand and R. J. Till, "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems," *Machine Learning*, vol. 45, no. 2, pp. 171–186, 2001. [Online]. Available: http://link.springer.com/10.1023/A:1010920819831

[79] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, jun 2006.

[80] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[81] F. J. Anscombe, "Graphs in statistical analysis," *The American Statistician*, vol. 27, no. 1, pp. 17–21, 1973. [Online]. Available: http://www.jstor.org/stable/2682899

[82] J. Matejka and G. Fitzmaurice, "Same Stats, Different Graphs," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. New York, New York, USA: ACM Press, 2017, pp. 1290–1294. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3025453.3025912

[83] Z. Wang, W. Yan, and T. Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline," *arXiv:1611.06455 [cs, stat]*, 2016. [Online]. Available: http://arxiv.org/abs/1611.06455{%}5Cnhttps://arxiv.org/abs/1611.06455

[84] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time Series," M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1998, ch. Convolutio, pp. 255–258. [Online]. Available: http://dl.acm.org/citation.cfm?id=303568.303704

[85] J. Masci, U. Meier, D. Cire\csan, and J. Schmidhuber, "Stacked Convolutional Auto-encoders for Hierarchical Feature Extraction," in *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I*, ser. ICANN'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 52–59. [Online]. Available: http://dl.acm.org/citation.cfm?id=2029556.2029563

[86] V. Turchenko, E. Chalmers, and A. Luczak, "A Deep Convolutional Auto-Encoder with Pooling - Unpooling Layers in Caffe," jan 2017. [Online]. Available: http://arxiv.org/abs/1701.04949

[87] H. Noh, S. Hong, and B. Han, "Learning Deconvolution Network for Semantic Segmentation," may 2015. [Online]. Available: http://arxiv.org/abs/1505.04366

[88] J. Dong, X.-J. Mao, C. Shen, and Y.-B. Yang, "Learning Deep Representations Using Convolutional Auto-encoders with Symmetric Skip Connections," nov 2016. [Online]. Available: http://arxiv.org/abs/1611.09119

[89] P. Bojanowski and A. Joulin, "Unsupervised Learning by Predicting Noise," *arXiv:1704.05310 [cs, stat]*, 2017. [Online]. Available: http://arxiv.org/abs/1704.05310

[90] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," feb 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[91] P. Usoro, S. Negahdaripour, I. Schick, and R. Nadira, "HVAC Systems Fault Diagnosis and Energy Optimization Using State Space Methods and Modern Control Theory," Tech. Rep., 1984.

[92] P. B. Usoro, S. Negahdaripour, and R. Nadira, "MODELING AND SIMULATION OF AN HVAC AIR HANDLER UNIT." mar 1985. [Online]. Available: https://miami.pure.elsevier.com/en/publications/modeling-and-simulation-of-an-hvac-air-handler-unit

[93] X. Zhou, "Dynamic modeling of chilled water cooling coils," Ph.D. dissertation, Purdue University, 2005. [Online]. Available: http://docs.lib.purdue.edu/dissertations/AAI3210824/

[94] P. Xu, P. Haves, and D. Curtil, "A library of HVAC component models for use in automated diagnostics," *IBPSA-USA Journal*, 2006. [Online]. Available: http://ibpsa-usa.org/index.php/ibpusa/article/view/223

[95] B. Tashtoush, M. Molhim, and M. Al-Rousan, "Dynamic model of an HVAC system for control analysis," *Energy*, vol. 30, no. 10, pp. 1729–1745, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544204004761

[96] J. Wang, C. Zhang, and Y. Jing, "Hybrid CMAC-PID Controller in Heating Ventilating and Air-Conditioning System," in *2007 International Conference*

on Mechatronics and Automation. IEEE, aug 2007, pp. 3706–3711. [Online]. Available: http://ieeexplore.ieee.org/document/4304163/

[97] R. Z. Homod, "Review on the HVAC System Modeling Types and the Shortcomings of Their Application," *Journal of Energy*, vol. 2013, pp. 1–10, 2013. [Online]. Available: http://www.hindawi.com/journals/jen/2013/768632/

[98] P. Li, H. Qiao, Y. Li, J. Seem, J. Winkler, and X. Li, "Recent advances in dynamic modeling of HVAC equipment. Part 1: Equipment modeling," *HVAC&R Research*, 2014. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10789669.2013.836877

[99] A. Afram and F. Janabi-Sharifi, "Review of modeling methods for HVAC systems," *Applied Thermal Engineering*, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359431114002348

[100] H. Li, D. Yu, and J. E. Braun, "A review of virtual sensing technology and application in building systems," *HVAC&R Research*, 2011.

[101] H. Li and J. Braun, "Virtual refrigerant pressure sensors for use in monitoring and fault diagnosis of vapor-compression equipment," *HVAC and R Research*, vol. 15, no. 3, pp. 597–616, 2009. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-77749338906{&}partnerID=40{&}md5=25dbbfd2c06823dd25ba5be1fa9d1511

[102] H. Li and J. E. Braun, "Development , Evaluation , and Demonstration of a virtual refrigernt charge sensor," *HVAC&R Research*, vol. 15, no. 1, pp. 117–136, 2009.

[103] S. Wang and J. Cui, "A Robust Fault Detection and Diagnosis Strategy for Centrifugal Chillers," *HVAC&R Research*, vol. 12, no. 3, 2006.

[104] P. Taylor, M. Street, L. Wt, S. Wang, and J. Cui, "A Robust Fault Detection and Diagnosis Strategy for Centrifugal Chillers A Robust Fault Detection and Diagnosis Strategy," *HVAC&R Research*, vol. 12, no. August 2012, pp. 37–41, 2011.

[105] G. Liu and M. Liu, "Development of a Pump Water Flow Station for HVAC Systems," in *ASME 2007 Energy Sustainability Conference*. ASME, 2007, pp. 633–637. [Online]. Available: http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1603207

[106] A. Wichman and J. E. Braun, "A smart mixed-air temperature sensor," *HVAC&R Research*, vol. 15, no. 1, pp. 101–115, 01 2009. [Online]. Available: http://search.proquest.com/docview/213366895?accountid=9902

[107] M. Yang and H. Li, "A virtual outside air ratio in packaged air conditioners," *HVAC&R Research*, 2011.

[108] M. . Ward and J. Siegel, "Modeling Filter Bypass: Impact on Filter Efficiency," *ASHRAE Transactions*, vol. 111, 2005.

[109] Z. Liao and A. Dexter, "A simplified physical model for estimating the average air temperature in multi-zone heating systems," *Building and environment*, vol. 39, no. 9, pp. 1013–1022, 2004.

[110] A. Wichman and J. E. Braun, "A Smart Mixed-Air Temperature Sensor," *HVAC&R Research*, 2009.

[111] X. Peng, P. Haves, and M. Kim, "Model-Based Automated Functional Testing – Methodology and Application to Air-Handling Units," *ASHRAE Transactions*, vol. 111, pp. 979–989, 2005. [Online]. Available: http://gaia.lbl.gov/btech/papers/55802.pdf

[112] A. Thosar, A. Patra, and S. Bhattacharyya, "Feedback linearization based control of a variable air volume air conditioning system for cooling applications," *ISA Transactions*, vol. 47, no. 3, pp. 339–349, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0019057808000190

[113] L. Ljung, "System identification," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[114] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte *et al.*, "Energyplus: creating a new-generation building energy simulation program," *Energy and buildings*, vol. 33, no. 4, pp. 319–331, 2001.

[115] C. Park, D. R. Clark, and G. E. Kelly, "An overview of hvacsim+, a dynamic building/hvac/control systems simulation program," in *Proceedings of the 1st Annual Building Energy Simulation Conference, Seattle, WA*, 1985, pp. 21–22.

[116] W. A. Beckman, L. Broman, A. Fiksel, S. A. Klein, E. Lindberg, M. Schuler, and J. Thornton, "Trnsys the most complete solar energy system modeling and simulation software," *Renewable energy*, vol. 5, no. 1-4, pp. 486–488, 1994.

[117] D. Brück, H. Elmqvist, S. E. Mattsson, and H. Olsson, "Dymola for multi-engineering modeling and simulation," in *Proceedings of modelica*, vol. 2002, 2002.

[118] P. Fritzson and V. Engelson, "ModelicaâĂŤa unified object-oriented language for system modeling and simulation," in *European Conference on Object-Oriented Programming*. Springer, 1998, pp. 67–90.

[119] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica buildings library," *Journal of Building Performance Simulation*, vol. 7, no. 4, pp. 253–270, 2014.

[120] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, M. Rosenberg *et al.*, "Us department of energy commercial reference building models of the national building stock," 2011.

[121] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

Appendix A

Identified BAS points

A.1 BAS points in AHUs

Table A.1: List of points related to the AHU

Group	Point Name	Description
Temperature	SAT	supply air temperature
	OAT	outside air temperature
	RAT	return air temperature
	MAT	mixed air temperature
	EAT	exhaust air temperature
	ZAT	air temperature of the zone supplied by AHU directly
	PHC AIR OutT	preheating coil leaving water temperature
	PHC WATER OutT	preheating coil leaving air temperature
	CLC AIR InT	cooling coil inlet air temperature (air temp before coil)

	CLC AIR OutT	cooling coil outlet air temperature (air temp after coil)
	CLC WATER InT	cooling coil inlet water temperature
	CLC WATER OutT	cooling coil outlet water temperature
	HTC AIR InT	heating coil inlet air temperature (air temp before coil)
	HTC AIR OutT	heating coil outlet air temperature (air temp after coil)
	HTC WATER InT	heating coil inlet water temperature
	HTC WATER OutT	heating coil outlet water temperature
Humidity	SAH	supply air humidity
	OAH	outside air humidity
	RAH	return air humidity
	MAH	mixed air humidity
	CLC AIR InH	cooling coil inlet air humidity (air temp before coil)
	CLC AIR OutH	cooling coil outlet air humidity (air temp after coil)
Flow	SAF	supply air flow rate
	RAF	return air flow rate
	OAF	outside air flow rate
	CLD F	cold duct air flow rate (dual duct AHU)
	HT F	hot duct air flow rate (dual duct AHU)
	HTC AF	heating coil air flow rate
	ChW F	chilled water flow rate

	HW F	hot water flow rate
Air Pressure	OA STATIC AP	outside air static air pressure
	RA STATIC AP	return air static air pressure
	SA STATIC AP	supply air static air pressure
CO2	SA CO2	supply air CO2 level
	EA CO2	exhaust air CO2 level
Occupancy	OCC	occupany (to decide whether the AHU system is in occupied mode or not)
Setpoint	SAT SP	supply air temperature setpoint
	RAT SP	return air temperature setpoint
	HT AIRT SP	hot duct air temperature setpoint (dual duct AHU)
	CLD AIRT SP	cold duct air temperature setpoint (dual duct AHU)
	ZAT SP	air temperature setpoint of the zone supplied by AHU directly
	MAT SP	mixed air temperature setpoint
	OAF SP	outside air flow rate setpoint
	RAF SP	return air flow rate setpoint
	SAF SP	supply air flow rate setpoint
	SA STATIC AP SP	supply air static air pressure setpoint
Valve	ChW VLV	chilled water valve position
	ChW VLV CMD	chilled water valve position control command
	HW VLV	hot water valvle position

	PHT VLV	preheat valve position
Damper	OAD	outside air damper position
	EAD	exhaust air damper position
	MAD	mixed air damper position
	RAD	return air damper position
Fan	FAN SPEED	fan speed of the AHU
	HR FAN SPEED	heat recover fan speed
	RA FAN SPEED CMD	return fan speed control command
	SF SPEED	supply air fan speed
	SF SPEED CMD	supply air fan speed control command
	SF STATUS	supply air fan status (ON/OFF)
	RF SPEED	return air fan speed
	RF SPEED CMD	return air fan speed control command
	RF STATUS	return air fan status (ON/OFF)
Electrical	ChW PUMP POWER	chilled water pump power
	FAN POWER	fan power
	PUMP POWER	pump power
	RF POWER	return air fan power
	SF POWER	supply air fan power
	UNIT CURRENT	unit current
	UNIT POWER	power consumption of whole unit
	UNIT VOLTAGE	unit voltage

A.2 BAS points in Terminal Boxes

Table A.2: List of points related to the terminal box

Category	Point Name	Description
Temperature	VAV POST RH T	air temperature after reheat valve in vav
	VAV PRE RH T	air temperature before reheat valve in vav
	VAV SAT	supply air temperature out of VAV
	VAV ZAT	zone air temperature supplied by VAV
	CAV SAT	supply air temperature out of CAV (constant air volume box)
Humidity	VAV ZAH	zone air humidity supplied by VAV
Flow	VAV SAF	vav supply air flow rate
	VAV ZAF	vav zone air flow rate
Air Pressure	VAV SA STATIC AP	vav supply air static pressure
	VAV ZAP	vav zone air pressure
Setpoint	VAV COOL SP	vav cooling temperature setpoint
	VAV HEAT SP	vav heating temperature setpoint
	VAV ZAT SP	vav zone air temperature setpoint
	VAV SAF SP	vav supply air flow rate setpoint
	VAV SAF MAX SP	supply air flow rate maximum setpoint
	VAV SAF MIN SP	supply air flow rate minimum setpoint
	VAV ZAF SP	zone air flow setpoint
	VAV ZAF MAX SP	zone air flow rate maximum setpoint
	VAV ZAF MIN SP	zone air flow rate minimum setpoint

	VAV SA STATIC AP SP	supply air static air pressure setpoint
Valve	VAV RH VLV	vav box reheat valve position
Damper	VAV DP	vav damper position

A.3 BAS points in RTUs

Table A.3: List of points related to the RTU

Category	Point Name	Description
Temperature	RTU RAT	return air temperature
	RTU EVP AIR InT	evaporator air inlet temperature
	RTU EVP AIR OutT	evaporator air outlet temperature
	RTU CLC WATER InT	cooling coil water inlet temperature
	RTU CLC WATER OutT	cooling coil water outlet temperature
Humidity	RTU RAH	return air humidity
Electrical	RTU FAN CURRENT	fan current
	RTU ACCONDENSER COMPRESSOR POWER	power consumption of the air conditioner compressor from an RTU

Appendix B

Supplement Materials of Large Scale Evaluation

B.1 Implementation Details

In this appendix, we specifically talk about the implementation details for feature extractions and the parameters for the classifiers. We mainly use **numpy**,**pandas**, **scikit-learn** packages for all implementations.

B.1.1 Data Cleaning

Since different sensing points have distinct sampling intervals ranging from one second to one hour, we re-sampled all the points to 15 minutes intervals using padding by filling values forward. Specifically, this was implemented using the **re-sample** function from the **pandas** package available for Python. A code snippet for the re-sampling process can be seen below.

```
def extract_for_one_customer(this_customer, start_t, end_t, freq =
    '15Min', debug = False):
```

```
base_time = pd.date_range(start_t,end_t,freq=freq)
ts_dim = len(base_time)

DataX = []
IgnoreX = []
Meta = []

for eq,pts in this_customer.items():
   for pt,val in pts.items():
      val = val.dropna().groupby(level=0).last()
      if val.size != 0:
         if debug:
            print([eq,pt])
         interpolated_data = val.resample(freq,
             fill_method='ffill')[start_t:end_t]
         if interpolated_data.size == ts_dim:
            DataX.append(interpolated_data.values)
            Meta.append([eq,pt])
         else:
            #skip the point if it does not have enough data
            IgnoreX.append([eq,pt,interpolated_data])
return DataX,Meta,IgnoreX
```

Additionally, we removed samples if they either had unclear descriptions or exhibited abnormal values. The specific rules for this are provided below:

- we remove the points if they are measuring humidity and contain negative values;

- we remove the points if they are measuring temperature and contain values smaller than -50 or greater than 300;
- we remove points if their description of point points is just "Point".

A code snippet for the data cleaning can be seen below.

```
# clean abnormal points
ix = [i for i in np.unique(np.where(X_raw < 0)[0]) \
      if 'Humidity' in pt_types[i]] +
     [i for i in np.unique(np.where(X_raw < -50)[0]) or \
      i in np.unique(np.where(X_raw > 300)[0]) \
      if 'Temperature' in pt_types[i]] +
     [i for i in range(len(y_raw)) \
      if 'Point' in pt_types[i]] # remove points named 'Point'
X = np.delete(X, ix, axis=0)
y = np.delete(y, ix, axis=0)
df = df.drop(ix)
```

B.1.2 Features

We implemented 6 different types of features as is seen in Table 3.1. Additionally, we combine all 6 features to generate the 7-th feature. The details of each feature are described as follows:

- For "F1: Li et al. 1994" [74], we extract mean, variance and coefficient of variation;
- For "F2: Gao et al. 2015" [42], in additional to what is described in the table, we include the 2-nd to 4-th order of central moments of the data, as well as the

entropy. The entropy is calculated by digitizing the data to 100 bins evenly if it contains more than 100 discrete values.

- For "F3: Hong et al. 2015" [41], we use the exact features described in the table.

- For "F4: Bhattacharya et al. 2015" [47], we use the exact features described in the table.

- For "F5: Balaji et al. 2015" [40], we also use 100 bins to digitize the data when calculating the entropy.

- For "F6: Koh et al. 2016" [49], we use the amplitude of the first three frequency components.

- For "F7: Combination", we simply combine all the previous features.

B.1.3 Classifiers

Seven classifiers are used, namely k-nearest neighbor (kNN), naive Bayes, logistic regression, linear discriminant analysis (LDA), decision tree, random forest, and AdaBoost. Both random forest and Adaboost use decision trees as the base classifiers to build the ensemble classifier. We vary some parameters of those classifiers, but we notice the performance is not significantly affected. We did also try SVM with RBF kernel. Due to the long running time and low performance, we did not include it in the results. For reference purpose, the following parameters are used for the classifiers:

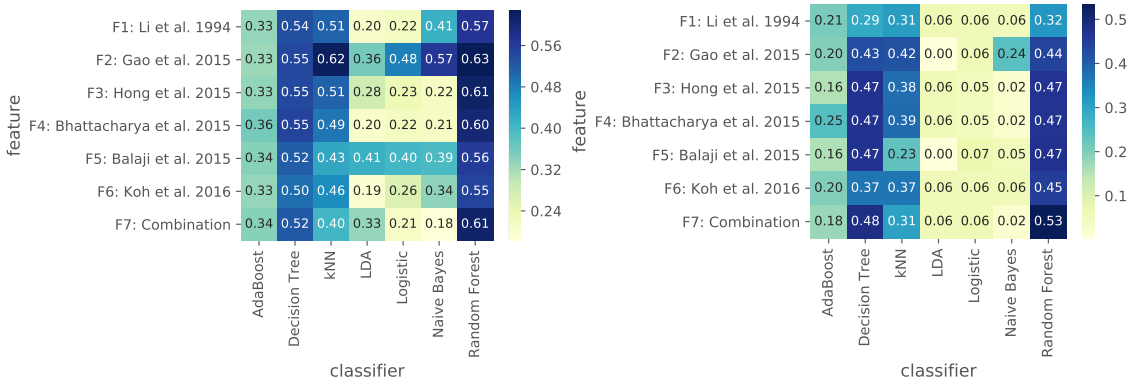
classifier	parameters
kNN	k=3
Logistic	C=1e5
Decision Tree	max depth = 10
Random Forest	max depth=10, number of estimators=20
AdaBoost	max depth=10, number of estimators=100

Table B.1: The parameters used for different classifiers

B.2 Performance of Other Metrics

B.2.1 Macro F_1 Score Matrix for Features and Classifiers

We show *macro* F_1 score matrices for both strategies in Figure B.1, which have the similar trend compared with accuracy score. However, the overall values are smaller compared with *micro* F_1 score (accuracy) due to a few classes with low performance decreases the overall *macro* F_1 score.



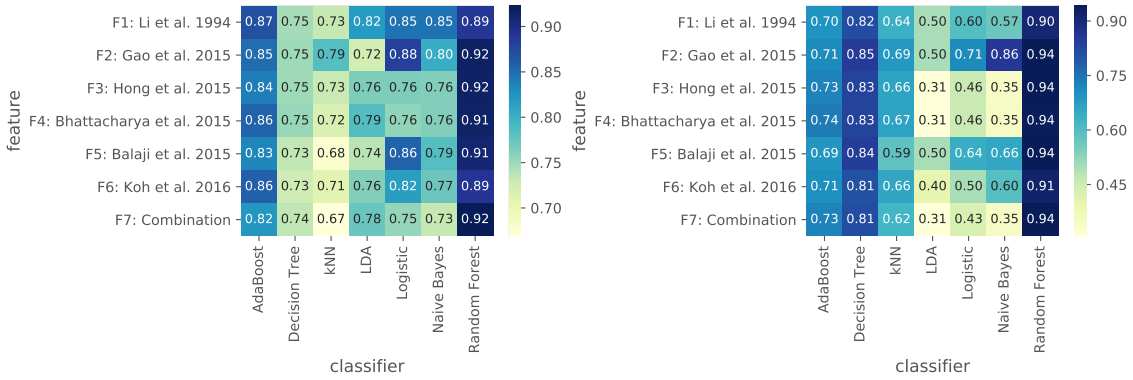
(a) macro F_1 score matrix (S1) (b) macro F_1 score matrix (S2)

Figure B.1: Macro F_1 score matrix from two strategies

B.2.2 Macro AUC Score Matrix for Features and Classifiers

We show macro AUC score matrices for both strategies in Figure B.2, which have the similar trend compared with accuracy score. Macro AUC is generated by "averaging" over individual AUC calculated based on a "one versus all" binary

classifier is built for each class. It shows a very high value, which is largely due to the number of true negatives is pretty high.

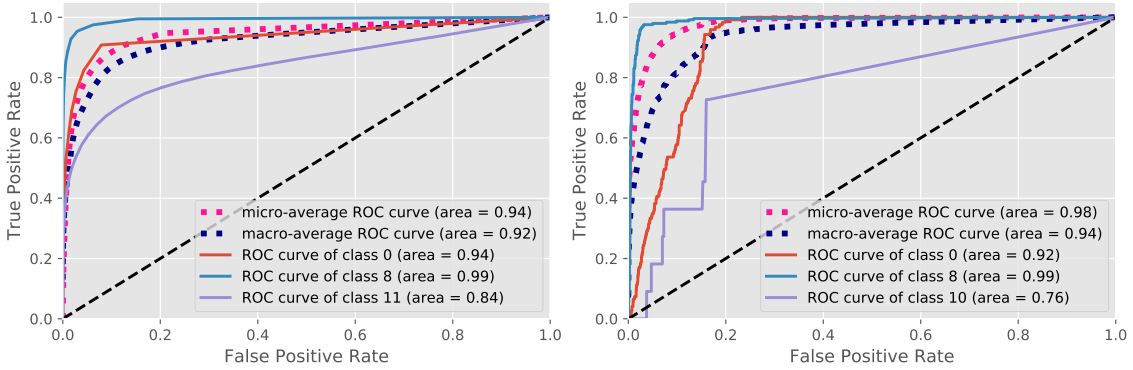


(a) macro AUC score matrix (S1) (b) macro AUC score matrix (S2)

Figure B.2: Macro AUC score matrix from two strategies

B.2.3 ROC Examples

We show the ROC examples using"F7: Combination" and "Random Forest" for both strategies in Figure B.3.



(a) ROC examples (S1) (b) ROC examples (S2)

Figure B.3: ROC examples from two strategies

B.2.4 Single Class Metrics for Each Class

We show the precision, recall, F_1 score, AUC and support using"F7: Combination" and "Random Forest" for both S1 and S2 in Table B.2 and Table B.3. The column "support" represents the ratio of samples for the corresponding class.

	precision	recall	F_1 score	AUC	support
AHU_Heating_Valve_Command	0.63	0.43	0.51	0.939	0.016
AHU_Cooling_Valve_Command	0.66	0.63	0.65	0.962	0.057
AHU_Mixed_Air_Temperature_Sensor	0.49	0.44	0.46	0.911	0.058
AHU_Outside_Air_Temperature_Sensor	0.89	0.90	0.90	0.987	0.043
AHU_Return_Air_Temperature_Sensor	0.57	0.53	0.55	0.938	0.063
AHU_Discharge_Air_Temperature_Sensor	0.63	0.68	0.65	0.878	0.089
AHU_Discharge_Air_Temperature_Setpoint	0.78	0.62	0.69	0.942	0.033
AHU_Outside_Air_Humidity_Sensor	0.96	0.87	0.91	0.991	0.016
AHU_Return_Air_Humidity_Sensor	0.88	0.81	0.84	0.992	0.037
AHU_Outside_Air_Damper_Position_Command	0.35	0.24	0.29	0.888	0.009
AHU_Mixed_Air_Damper_Position_Command	0.11	0.01	0.01	0.804	0.002
Other	0.85	0.89	0.87	0.843	0.578

Table B.2: Precision, recall, F_1 score, AUC and support for each class (S1)

	precision	recall	F_1 score	AUC	support
AHU_Heating_Valve_Command	0.04	0.01	0.01	0.916	0.020
AHU_Cooling_Valve_Command	0.51	0.52	0.51	0.956	0.057
AHU_Mixed_Air_Temperature_Sensor	0.48	0.49	0.48	0.936	0.057
AHU_Outside_Air_Temperature_Sensor	0.86	0.68	0.76	0.976	0.044
AHU_Return_Air_Temperature_Sensor	0.51	0.67	0.58	0.960	0.062
AHU_Discharge_Air_Temperature_Sensor	0.68	0.70	0.69	0.968	0.086
AHU_Discharge_Air_Temperature_Setpoint	0.89	0.86	0.87	0.980	0.039
AHU_Outside_Air_Humidity_Sensor	0.98	0.81	0.89	0.994	0.018
AHU_Return_Air_Humidity_Sensor	0.76	0.74	0.75	0.989	0.040
AHU_Outside_Air_Damper_Position_Command	0.00	0.00	0.00	0.925	0.015
AHU_Mixed_Air_Damper_Position_Command	0.00	0.00	0.00	0.763	0.002
Other	0.84	0.87	0.85	0.916	0.561

Table B.3: Precision, recall, F_1 score, AUC and support for each class (S2)

Appendix C

Supplement Materials of CNN Approach

C.1 Implementation Details of Baseline Features

In this appendix, we specifically talk about the implementation details for four feature extractions. We mainly use **numpy**,**pandas**, **scikit-learn** packages for all implementations. Additionally, we combine all four features to generate the fifth combined feature(**combF**). The details of each feature are described as follows:

- **statF**: The following statistical quantities are included namely minimum, median, mean, maximum, standard deviation, skewness, kurtosis, entropy (100 bins are used to digitize the data), 2/9/25/75/91/98-th percentiles, mode, and coefficient of variation. Additionally, the signal energy, the slope, the first and second variance of the difference between consecutive samples, the number of up and down changes are also included.

- **winF**: We calculate 22 statistical features (**statF**) on N sliding windows of length 4 with an overlapping of 2. Then another set of statistics including min-

imum, median, maximum, and standard deviation are used over N windows, which eventually produces the feature vector of length 88.

- **tfaF**: The fifth level of detailed wavelet coefficients based on "Harr" wavelets are used. The amplitude of top 20 frequent components based on FFT is used as well.

- **dtwF**: The Euclidean distance is used as the distance metric to calculate the warping distance. Once calculated, a log scale will be applied to the original calculated dynamic warping distance.

C.2 Performance of Other Metrics fo CAE

We show the precision, recall, F_1 score, AUC and support using "caeF" and "Random Forest" in Table C.1. The column "support" represent the ratio of samples for the corresponding class.

	precision	recall	F_1 score	AUC	support
PreheatTemperature	0.17	0.01	0.02	0.814	0.022
OutdoorAirFlow	0.68	0.64	0.66	0.973	0.030
OutdoorAirHumidity	0.91	0.78	0.84	0.985	0.023
SupplyFanCommand	0.36	0.07	0.12	0.951	0.043
MixedAirTemperature	0.41	0.37	0.39	0.896	0.074
CoolingOutput	0.59	0.76	0.67	0.969	0.073
OutsideAirTemperature	0.94	0.75	0.84	0.968	0.056
Occupancy	0.23	0.11	0.15	0.936	0.027
SupplyFanStatus	0.48	0.83	0.60	0.957	0.065
DischargeAirTemperature	0.55	0.75	0.63	0.941	0.110
ReturnAirHumidity	0.86	0.95	0.90	0.995	0.051
SupplyFanOutput	0.63	0.67	0.65	0.935	0.065
DischargeAirFlow	0.69	0.69	0.69	0.979	0.025
ReturnFanOutput	0.06	0.02	0.03	0.842	0.023
DuctStaticPressure	0.97	0.97	0.97	0.991	0.068
ReturnAirTemperature	0.44	0.55	0.49	0.935	0.079
ZoneTemperature	0.49	0.35	0.41	0.953	0.052
ReturnAirQuality	0.95	0.94	0.94	0.991	0.040
DischargeAirTemperatureSetpoint	0.79	0.76	0.77	0.984	0.049
HeatingOutput	0.17	0.07	0.10	0.919	0.026

Table C.1: Precision, recall, F_1 score, AUC and support for each class