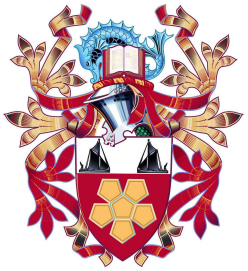


London South Bank University

Data-driven remote fault detection and diagnosis of HVAC terminal units using machine learning techniques.

by

Maitreyee Dey

A thesis presented in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Engineering
Department of Electrical & Electronics Engineering
London South Bank University
London, United Kingdom

October 2019

Data-driven remote fault detection and diagnosis of HVAC terminal units using machine learning techniques.

Abstract

The modernising and retrofitting of older buildings has created a drive to install building management systems (BMS) aimed to assist building managers pave the way towards smarter energy use, improve maintenance and increase occupants comfort inside a building. BMS is a computerised control system that controls and monitors a building's equipment, services such as lighting, ventilation, power systems, fire and security systems, etc. Buildings are becoming more and more complex environments and energy consumption has globally increased to 40% in the past decades. Still, there is no generalised solution or standardisation method available to maintain and handle a building's energy consumption. Thus this research aims to discover an intelligent solution for the building's electrical and mechanical units that consume the most power. Indeed, remote control and monitoring of Heating, Ventilation and Air-Conditioning (HVAC) units based on the received information through the thousands of sensors and actuators, is a crucial task in BMS. Thus, it is a foremost task to identify faulty units automatically to optimise running and energy usage. Therefore, a comprehensive analysis on HVAC data and the development of computational intelligent methods for automatic fault detection and diagnosis is been presented here for a period of July 2015 to October 2015 on a real commercial building in London. This study mainly investigated one of the HVAC sub-units namely Fan-coil unit's terminal unit (TU). It comprises of the three stages: data collection, pre-processing, and machine learning. Further to the aspects of machine learning algorithms for TU behaviour identification by employing unsupervised, supervised, and semi-supervised learning algorithms and their combination was employed to make an automatic intelligent solution for building services. The accuracy of these employed algorithms have been measured in both training and testing phases, results compared with different suitable algorithms, and validated through statistical measures. This research provides an intelligent solution for the real time prediction through the development of an effective automatic fault detection and diagnosis system creating a smarter way to handle the BMS data for energy optimisation.

Declaration

The work in this thesis is based on research carried out at the Biomedical Engineering and Communications (BiMEC) Research Centre, School of Engineering, London South Bank University, London, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgments

First and foremost, I would like to express all my gratitude to Prof. Sandra Dudley for believing in me and giving me the opportunity to pursue a PhD degree. Thank you for your patience and continuous support throughout these years we spent together.

I would also like to thanks to Demand Logic team for providing me the invaluable information about the building data, verification and feedback on building services, and related performance issues. Furthermore, I would like to thank the London South Bank University and Demand Logic Ltd. for funding my PhD studies as a part of the Innovate UK funded research project (EP/M506734/1) and providing me the highest-level facilities and resources to perform this research.

I want to thank my friend Soumya for giving me a greatest support while needed throughout this PhD Journey. Also, want to thank Mahreen and all the colleagues of the Biomedical Engineering and Communications (BiMEC) Research Centre of LSBU.

My warmest thanks go to my Husband. Your continuous support and love were one of the key drivers for me not to give up in harsh times. My gratefulness goes to my father and mother for their continuous love, support, and belief. All of you have been my role models and I would never be the person that I am today without you. This thesis would not be possible without your support and I dedicate this thesis to all of you.

Contents

List of Figures

List of Tables

Nomenclature

μ_d	mean difference
A_E	Area of an event
C	Cooling
F_{C_k}	Feature of cooling area
F_{H_k}	Feature of heating area
H	Heating
H_0	Null hypothesis
Pr	Precision
Re	Recall
BAS	Building automation system
BIC	Bayesian information criterion
BMS	Building management system
DAD	data acquisition device
DB	Davies-Bouldin indexing
EA	Event aggregation
EAC	Event area calculation
ED	Event discovery
EE	Event end
ES	Event start

FCU Fan coil unit

FN False negative

FP False positive

GA Goal achieved

HVAC Heating, ventilation, and air-conditioning

KNN K-nearest neighbour

LMC-SVM Linear multi-class support vector machine

MC-SVM Multi-class support vector machine

MLP Multilayer perceptron

PID Proportional integral derivative

QMC-SVM Quadratic multi-class support vector machine

RD Response delay

SI Silhouette indexing

SL Supervised learning

SSL Semi-supervised learning

TP True positive

TU Terminal unit

UL Unsupervised learning

Chapter 1

Introduction

> "Arise, awake, and stop not until the goal is achieved."
>
> *-Swami Vivekananda*

Buildings are becoming increasingly complex energy consuming systems, comprising several elements such as heating, ventilation, and air-conditioning (HVAC) systems, lighting systems, security systems, fire alarms, lifts, elevators, closed-circuit television, and so on. Thus, new buildings are being created with, whilst older buildings are being retrofitted with building management systems (BMS), also known as building automation systems (BAS). The BMS is a computerized control system that controls and monitors building services, plants, components, devices such as sensors, actuators, valves, dampers, motors, pump, fans, etc. BMS is defined as "an IT-based solution that extends the capabilities of sensing, control, and automation hardware to direct automated and/or manual improvements to system operations utilizing the data from multiple streams [3]. It is a rapidly growing market with BMS revenues for hardware, software and services projected to rise from today's $2.7 billion worldwide to $12.8 billion by 2025 [4]. Thus,

monitoring building energy, financial reward, and concern for the environment has been the focus in recent years.

1.1 Benefits and Pitfalls of BMS

Ideally, a BMS assists building managers to understand how their building is functioning and permits managers to control and adjust the system to optimize performance. A BMS collects data, can help to visualize it, generate reports, create alarms when system failures occur. The BMS aims to manage several units and their energy demands in an effective way, leading to good quality, informed decision making, improved performance, and a reduction of energy wastage. A BMS primarily facilitates good control of individual room temperature, effective and remote monitoring of building units, targeting energy consumption, improve plant reliability and life, maintain occupant comforts, computerized scheduling, effective responses to HVAC-related complaints and provides useful information related to these assistances. However, due to the manual involvement or inexperienced technicians, it can lead to negating effective performance monitoring of BMS resulting in delayed on forgotten actions taken not been carried out on broken units, leading to deteriorating equipment, discomfort, energy wastage, etc. Thus, efficient and effective monitoring of BMS data is a pivotal research area.

Recent research has found that building sectors account for 40% of global energy consumption [5]. Additionally, in residential and commercial buildings, HVAC systems alone are responsible for approximately 39% - 41% of the total building energy consumption [5]. Therefore, the HVAC is an important, as well as complex system comprising several units such as air handling unit (AHU), variable air volume (VAV), chiller, boiler, fan coil unit (FCU) and so on. Commercial buildings generally have one or two major units i.e., a chiller or boiler but will have hundreds to thousands of small units i.e. fan coil units (depending on the size of building)

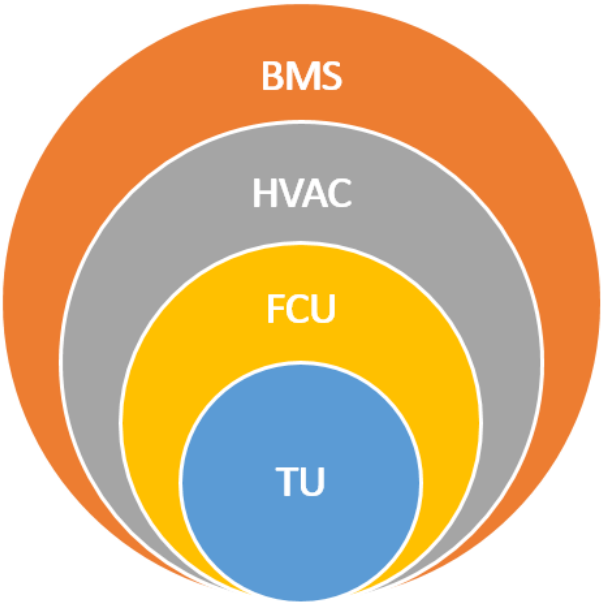


FIGURE 1.1: The overlapping relationship diagram of the building's terminal unit within BMS.

have fitted inside the building. Thus, if a single small unit is not working properly, it can affect the entire plant tremendously. To accommodate the BMS research need, this thesis has concentrated on a small but important unit: fan-coil unit's terminal unit (TU). Figure 1.1 shows the overlapping relationship from terminal unit to BMS. What is demonstrated is that the TU is part of the FCU, FCU is within the HVAC and the HVAC is part of the BMS. The description of these units is described in the next sections.

1.2 HVAC system

A building's HVAC system is integrated with three key functions: heating, ventilation and air-conditioning to deliver thermal comfort and provide satisfactory indoor air quality within acceptable installation, action, maintenance costs, and maintains the overall building's heating and cooling performance. HVAC systems are connected to the BMS and comprise several sub-systems such as a fan coil unit (FCU), air-handling unit (AHU), variable air volume (VAV), boiler, chiller, etc,

which interact with each other. Figure 1.2, displays the overall process flow from data collection through a platform from different building components to the final feedback generation (detailed in Chapter 3). Henceforth, this work investigated and analysed the behaviour of a specific HVAC sub-unit detailed next.

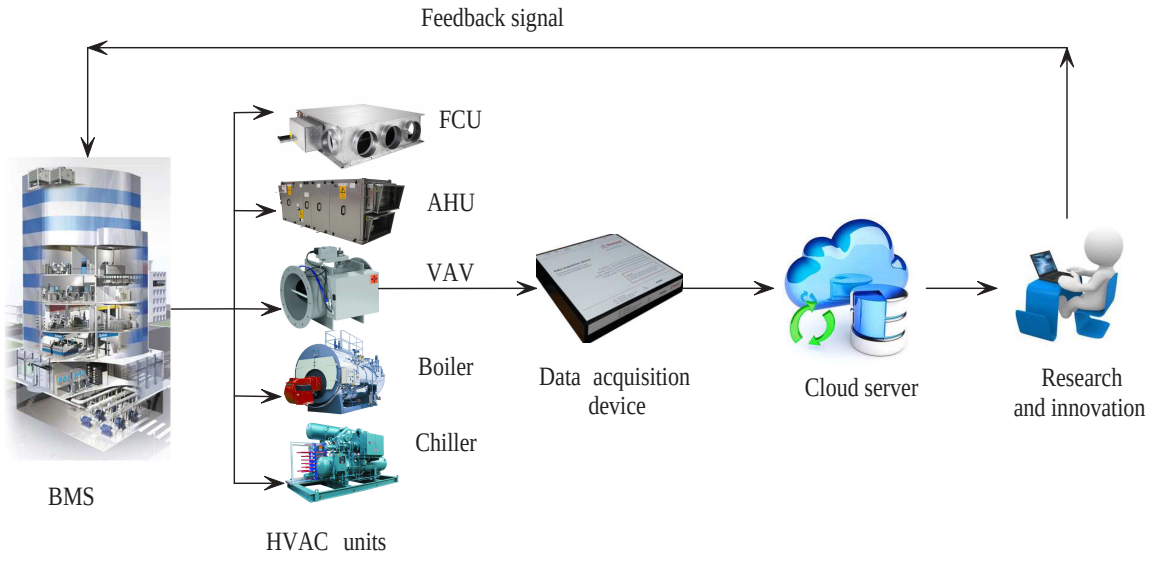


FIGURE 1.2: The general architecture of BMS system and the relation with HVAC units.

1.3 Fan coil unit's terminal unit

A fan coil unit is a specific sub-component or sub-unit of a HVAC system found in residential, commercial, and industrial buildings. A FCU consists of a heating coil, cooling coil and a fan or a damper. It is normally ceiling-mounted and controlled by local thermostats. Depending upon the individual room's thermostats the return air temperature may either recirculate internal air or introduce fresh air along with re-circulated air and discharge fresh air to the room. A typical FCU schematic is shown in Figure 1.3. Generally, inside a building, a central chiller and boiler plant distributes cold water to all the cooling coils and hot water to all the heating coils. If the environment becomes too warm (based on the set

point temperature or upon user preferred temperature requests inside a building), the thermostat senses the rise of temperature and signals the chilled water valve to flow, with cold water then passed through the cooling coil and the cool air being blown by the fan. If the room temperature becomes too cold (depending on the local set point or a user preferred temperature setting), the heating coil starts working in the same way and blows the hot air until the room temperature reaches the desired level or set-point.

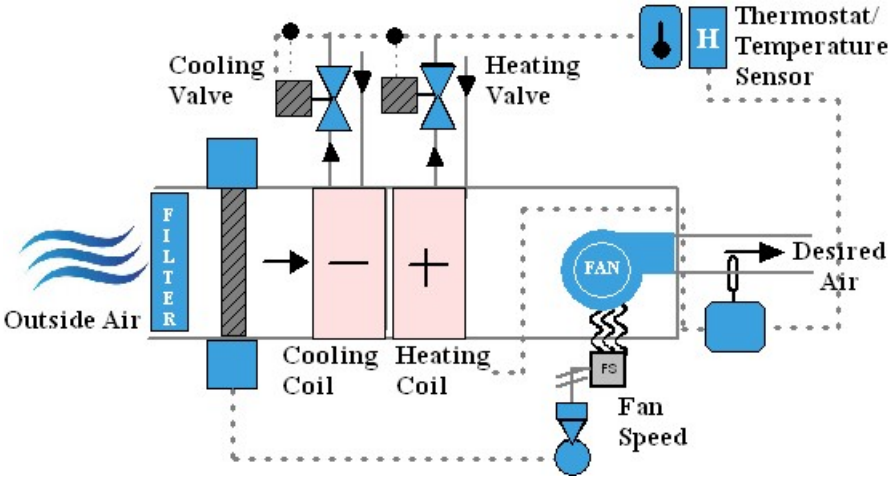


FIGURE 1.3: A schematic diagram of fan coil unit.

The FCU is an air terminal unit, and such units are popular in larger buildings. The overall heating and/or cooling air inside the rooms is distributed by the TU. Thus, the TU is responsible for the final delivery of comfort inside a built environment with obvious implications for energy consumption. Here, the research aims to analyse the behaviour of FCU's TU and it is represented as TU and sometimes also used interchangeably with FCU throughout the thesis. This research has closely involved a building management company "Demand Logic (DL) Ltd.", who informed the researcher of the research problem from a buildings engineer's perspective and engaged on determining the efficiencies of the results. Overall creating an building engineers insight into the advantage of applied machine learning outcomes in real life scenarios.

1.3.1 Terminal unit's parameters

The TU data gathers important information related to a building's behaviour. A single TU consists of various data streams also known as data parameters which are considered for this study. There are five parameters: control temperature, set-point, deadband, heating power, cooling power, and enable signal are the most significant to represents a single TU behaviours. Thus, these data streams have been collected by a sensor device (discussed in Chapter 3) implemented by the company (the research partner) "Demand Logic". The definitions of these parameters are as follows:

- Control temperature [°C] is measured at a fixed time interval (10 min) by each TU or in some cases a zone space temperature is used. These are generally measured by local thermostats/ sensors in the room.
- Set point temperature [°C] is the desired or target control temperature of the unit which is set by the operator or the administrator based on the current demand or depending upon the weather.
- Deadband is the control temperature band or range within the process that permits a range within which the system tries to keep the requested temperature and creates two separate output ranges of heating and cooling set point.
- Heating and cooling valve or damper actuator control and feedback signals, which provides the corresponding power demand.
- Enabled signal to indicate the hours of operation of a unit.

A single poorly controlled TU can be responsible for significant energy wastage and occupant discomfort. For example, a defective unit can signal a false heating demand to the boiler, causing the ancillary equipment to activate and begin

distributing hot water causing rooms to overheat. In the following subsection, possible TU issues are explained which display different abnormal behaviour (where the deviation is found between the set-point and measured value of variables) defined as 'fault'.

1.3.2 Terminal unit's issues

There are a number of potential issues that can be identified from TU data analysis. Various behavioural metrics such as saturation, on-ness and hunting have been studied for different TUs to identify the prevailing issues.

(a) Saturation can be defined as the proportion of time over a day that the valve or damper is open to its maximum. Thus the higher the value, the longer a heating or cooling valve (or damper) is open.

(b) On-ness can be defined as the proportion of time that a terminal unit has any heating or cooling demand over a 24-hour period (i.e. any time that demand is greater than 0%). Simply the period something is on.

(c) Hunting is calculated using the set point (mentioned above), control temperature and is a measure of how much this fluctuates over a day.

Even though a TU is considered a "simple device" there can be a multitude of issues that lead to faulty or abnormal behaviour and require expert building engineering knowledge through visualisation, experience, guess work, etc. to identify and conclude the cause of each one of these issues. Through experience some of the recognized possible causes that can result in faulty or abnormal behaviours are listed below:

(a) Poor control- Typically, this can happen due to too narrow dead bands boundaries being set and/or over aggressive proportional, integral and derivative

(PID) control. This can result in the TU frequently switching between heating and cooling.

(b) Poor sensor location- The temperature sensor could be located at a wrong position. For an instance, in close proximity to heating and/or cooling elements (e.g. back of a drinks cabinet).

(c) Varying set point- The set point may be varied too frequently, typically due to occupant discomfort (e.g. user is changing set point for personal preference).

(d) Out of hours operation- A unit may be found to operate out of hours because it has simply been forgotten (manually operated and left it on) or the operational time schedule is incorrect.

(e) Incorrect TU sizing for actual demand- It can happen that the load is underestimated and a bigger unit should have been installed. This is found more often in cooling than heating mode.

(f) TU unable to receive adequate flow or upstream temperatures- The flow temperature from the boiler or chiller is not sufficiently high or low to condition the space, sometimes due to over-ambitious temperature compensation.

(g) Stuck-open valve- Often indicated by a saturation value of 1, it could mean the valve was fully open over a 24-hour period.

(h) Competition from nearby TUs- A TU is trying to heat the space but an adjacent TU near or in the same space, is trying to cool the same area. This is generally found where there is poor hierarchical control over a branch of TUs.

(i) Localised effects- This can be caused either due to high solar gains or TU placed very close to energy-consumption equipment with high internal gains, like an old lighting fixture or photocopier.

(j) Unachievable set point- Sometimes it can happen that a user adjusts the set-point temperature to maximum or minimum value that is simply unachievable for that environment.

An example of faulty or abnormal and normal or non-faulty TU behaviour in terms of their control temperature is shown in Figure 1.4. It can be seen that the normal or non-faulty pattern is quite stable whereas the faulty one is volatile in nature. The graph of control temperature has been shown for a whole day (17^{th} July 2015) for one faulty and one non-faulty TU. The x-axis represents the time staring with midnight to 11 pm on that specified day with a time interval of 10 minutes. The y-axis represent the specific room temperature in a room of a particular building (detailed in the case study Section 3.3).

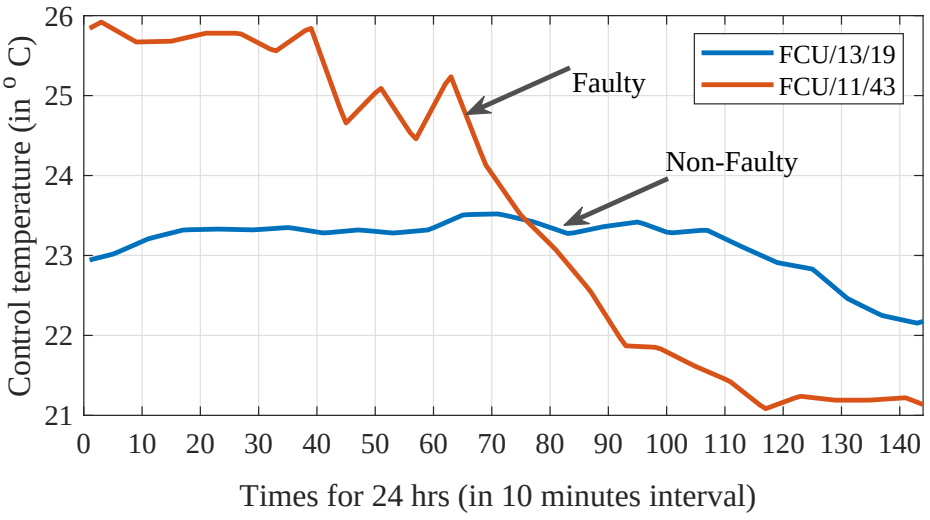


FIGURE 1.4: An example control temperature of faulty and non-faulty TU on a specific day.

Equipment failure and performance degradation of such units in commercial buildings often goes unnoticed until it causes an impact on occupant comfort, triggers an equipment-level alarm, deteriorates equipment life, all resulting in excessive energy consumption. Therefore, early detection of an unexpected behaviour and subsequent best practice on remedies can assist with these mentioned issues. All of which are currently monitored manually by expert building engineers or service

providers, who managed the BMS. Thus, monitoring TU performance to auto-identify unusual behaviour is the focus of this thesis.

1.4 Fault detection and diagnosis

Fault detection and diagnostics (FDD) is a process to identify anomalous behaviour in the performance of any BMS equipment i.e. terminal unit (TU). Fault detection aims to recognize a problem that has occurred even if the cause is unknown and fault diagnosis to indicate the causes of these problems, to the point where corrective action can be taken. Here, the FDD study began by analysing TU performance to recognise process problems, e.g. temperature, flows, pressures, level, power, control signals, etc. Specifically, it is an examination of a TU, which is not working desirably and unable to maintain the comfortable room temperature. Technically it can be said that even though the demand response (power) is high still the control strategy (temperature) is poor and as a result demands excess power, deteriorates equipment life and impacts occupant comfort.

For example, poor temperature control and unachievable set points due to narrow dead band settings (e.g. the heating and cooling set-point is set between 20^o C - 21^o C and whenever the room temperature goes beyond any of these limits it demands respective heating or cooling power depends on the rise or down) results in TU behaviour termed 'hunting'. In an office if the TU is left on beyond working hours this is a type of scheduling error results in unnecessary electricity use with additional power demands needed to maintain the room temperature even at night when workers are gone home. Room location is also a factor for abnormal behaviour (e.g. 17^o C ambient temperature requests in a room that has continuous sunshine exposure, or is settled at 26^o C ambient temperature in a summer day where the sensor is fixed near to a cooler room). All these situations can make FDD complex. Thus, it's obvious that there can still be a multitude of issues

leading to 'faulty' TU behaviour that require expert building engineer knowledge to identify each one of these issues as they occur or identify that malfunctions observed are the summation of individual faults.

The manual TU data investigation and fault-finding process is complex, time consuming, requires expert knowledge, expensive and can lead to losing a grip on problems within a building. Hence bringing automation and intelligence to this process using data mining and machine learning approaches could be an ideal solution. The large amount of data is the most powerful component and provides a foundation towards appropriate decision making. Thus, automation in the fault detection and diagnosis study relies heavily on input from sensors or derived measures of performance and defined as automatic fault detection and diagnosis (AFDD). The term FDD and AFDD has been used interchangeably throughout the thesis.

1.5 Machine learning approaches

Machine learning (ML) is an evolution of artificial intelligence (AI). ML provides the intelligence to machines to "learn" how to perform a specific set of tasks by processing data. For example, the scheduling services in BMS are generally handled by a control engineer or by building operators. The building scheduling has been based on some previous available data to make scheduling decisions automatic such as temperature, power demand, occupancy, and holidays/work week hours, and so on. This could be done by an experienced professional and can also be executed automatically, but no matter how efficiently it would be made, any human's ability is limited by the volume of data they can process to keep on top of issues. Whereas, machines can process millions of data points from thousands of sources. Therefore, machines with suitable algorithms have the potential to automatically discover devices which are 'misbehaving' and making

excessive demands on the major plants within a building and not possible to make decisions based on the complex data by humans.

1.6 Objectives of the thesis

Remote observation and fault identification makes an excellent contribution to successful BMS in terms of creating future augmentable systems for automatic problem solving. This approach effectively leads to the empowerment of building managers who are not always experienced in systems within the buildings they manage. Remote problem identification and tracking can not only lead to energy wastage reduction, but also provides further benefits such as increased operational cost savings and a greater appreciation and understanding of human-building interaction for future smart city applications and future building developments. In the multi-level and multiple areas of building research, many researchers have contributed research findings/thoughts to solve fault finding issues for different units which could lead to reduction of energy wastage. Real BMS data automatic fault identification is difficult due to the complex nature of the data. This generally involves an expert building engineers knowledge to interpret the detected faults of a unit. It is found from recent development of automatic fault detection and diagnosis literature that machine learning is effective in identifying faults automatically, however, it requires historic data and appropriate method to learn accordingly [5]. Though, there is no standard yet available for energy management in buildings research.

This PhD research being carried out as a part of a match funded PhD project in collaboration with Demand Logic and London South Bank University. Here, the use of machine learning to highly applied topics focuses specifically on remote analysis of real BMS data from buildings in London, UK. A novel data-driven feature

extraction method is proposed to represent the high dimensional data into a meaningful low dimensional data steams. The feature extraction method is performed by discovering events augmenting the proportional integral derivative (PID) controller response curve. Subsequently, the machine learning method is carried out to classify the 'faulty' and 'non-faulty' TU behaviour to assist building managers for fast diagnosis of issues in buildings. The investigation began by analysing historic data from a small but important on pivotal HVAC sub-component i.e. the terminal unit. The algorithm concentrates on data-driven based approaches, which could eventually classify faults from the different units (fan coil unit, variable air volume, air handling unit, chiller, boiler, etc.) regardless of faults. Thus, it can be used not only to detect and diagnose equipment failures, but also to provide significant energy savings and well-being impact through preemptive maintenance, behaviour analysis and predictive building identification.

1.7 Thesis contribution

This thesis involves access to real world data and collaboration with building engineers. A rigorous study on the data parameters, subsequent analysis, feature extraction, for faulty and non-faulty patterns of individual units is ensured. Theoretical development of the methodology for detection and diagnosis, investigation on how to classify faults automatically and perform predictive analysis on building data is all performed are presented. ML based methods have been employed to identify faults automatically and predict future scenarios to anticipate faults in real time. The entire process has been performed incorporating three learning stages as follows:

(a) Unsupervised learning

(b) Supervised learning

(c) Semi-supervised learning

In an unsupervised learning system, the data is unlabelled and unclassified, where the corresponding output is unknown. The machine must infer the relevance of the data, since BMS routinely uses complex data to establish system settings. Unsupervised learning investigates hidden structures in unlabelled data, leading to one of the most difficult tasks in ML theory. Therefore, a clustering algorithm has been performed on the HVAC data to identify distinct types of group of patterns. Thereafter these groups have been used to create pseudo labels on the historical data set for supervised learning. For supervised machine learning, all the data elements in the model need to be labelled, so the machine has instructions on how to classify the data. Thereafter, the semi-supervised learning method has been employed, which can make use of small amount of labelled data with a large amount of unlabelled data and automate the whole AFDD process. It falls between unsupervised and supervised learning. The purpose of using this is to make as much use of the available historic data and employ labelling processes to reduce the computational complexity by the labelling process once instead of labelling new data every time, making it suitable for real-time processing.

1.7.1 Thesis related publications

1. **M. Dey**, S. P. Rana, S. Dudley. "Smart building creation in large scale HVAC environments through automated fault detection and diagnosis", Future Generation Computer Systems, Elsevier, 2018. ***(Impact factor-5.768)***

2. **M. Dey**, S. P. Rana, S. Dudley. "Semi-Supervised Learning Techniques for Automated Fault Detection and Diagnosis of HVAC Systems", 30th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2018), Volos, Greece, 2018.

3. **M. Dey**, M. Gupta, S. P. Rana, M. Turkey, S. Dudley. "A PID Inspired Feature Extraction for HVAC Terminal Units" , IEEE conference on sustainablity and technology (SusTech), Phoenix, AZ, USA, 2017.

4. **M. Dey**, M. Gupta, M. Turkey, S. Dudley. "Unsupervised Learning Techniques for HVAC Terminal Unit Behaviour Analysis", IEEE International Conference on Smart City Innovations (SCI'17), San Francisco, USA, 2017.

5. "Human-Building Interaction Employing Neural Network for HVAC TU's Energy Optimisation: A Case Study Based Approach", Building and Environment, Elsevier, **(In preparation)**.

1.8 Thesis outline

The thesis is arranged as follows: The report begins with a literature review on the field of fault detection and diagnosis for HVAC systems, recent advancements and different strategy based approaches are presented in Chapter 2. A detailed description of the system architecture and terminal unit data collection, potential terminal unit issues, the developed big data framework, and the case studies are presented in Chapter 3. Following TU data collection, a new feature extraction method is proposed in Chapter 4. A proportional-integral-derivative based event discovery process has been developed to illustrate the theory of feature extraction method is provided in this chapter. Subsequently, unsupervised learning (UL) algorithms are employed on the extracted featured data and explained in Chapter 5. Three different algorithms have been explored and compared with all obtained results via statistical measures. Also, a radar graph based visualization method of faulty and non-fault patterns have been displayed to provide the effectiveness of the feature extraction along with the clustering for categorization of the distinct patterns and for labelling all the TU data. Further, supervised learning (SL) has been performed

on these labelled TU data to identify the specific faults and TU data classification has been presented in Chapter 6. Three types of classification algorithm have investigated, compared, and validated through well-established statistical measures for TU fault identification for a case study. Thereafter, semi-supervised learning (SSL) algorithm has been performed which is a class of machine learning technique which makes use of a small amount of labelled data with a large amount of unlabelled data and is demonstrated in Chapter 7. As in real life scenarios, ever increasing building data processing is highly complex, thus, SSL is employed here to make maximum use of the small amount of labelled training data and to deal with the huge levels of unlabelled TU data to test the building performance. This method was eventually able to predict faults in real-time through learning from the small amount of historical data. The algorithm performance has been validated through the statistical t-test and p-value measurements. Finally the main contribution of this thesis are summarized in Chapter 8. Some concluding remarks of this research and future research directions for this BMS research are also presented.

Chapter 2

Background

"Learn everything that is good from others but bring it in, and in your own way absorb it; do not become others."

-Swami Vivekananda

2.1 Introduction

Successful fault detection and diagnosis in buildings is a crucial area and a global problem in the field of energy management [6]. Worldwide, these energy related problems are becoming more prominent in the building sector as more complex building needs arise [7]. Although, huge interest and trust is applied to a building management system (BMS) there are no global specific standards or certificates available for energy management in BMS. Recently, the International Organization for Standardization (ISO) 50001:2018 began to integrate a framework of requirements for organizations to develop policies for more efficient energy usage but it is not obligatory to enforce or follow [8]. Building's energy systems are complicated and are affected by different factors such as building modelling structure,

outside weather conditions, occupant behaviour and so on. All of these factors have an impact on the different building components i.e. heating, ventilation and air-conditioning (HVAC) systems. Global research focuses on energy modelling for different HVAC systems to develop control strategies that would primarily result in an overall reduction of a building's energy consumption.

Although in building environments energy usage is accountable for the building space, lighting system, water heating, HVAC systems etc., it is found that most of the building's energy is consumed by the HVAC system [9]. Figure 2.1 summarises, via a pie chart, a commercial building's global energy consumption strategy, where it is noticed that approximately 41% energy is consumed by HVAC system [10, 11]. Thus, if the HVAC is under-performing this is where the greatest losses or gains can be found. From the literature it is also found that there are several opportunities to increase energy efficiency in newly constructed buildings which can decrease electricity consumption between 20% and 50% by including proper design in the room space, HVAC (20% - 60%), lighting (20% - 50%), water heating (20% - 70%), refrigeration (20% - 70%) and others (e.g., office based equipment and smart controls, 10% - 20%) [12].

Several developed fault finding approaches in various HVAC units have been discussed including engineering models, statistical models, machine learning models leading to energy restoration and diminishing environmental impact [13, 14]. In this chapter, FDD research over the past two decades is reviewed and presented. It provides an overall scenario for the already developed FDD methods and finds new research space in this domain. This study began by focusing on related review work by researchers in [15, 16, 17]. These state-of-the-art articles have demonstrated the important aspects of HVAC system throughout the UK, and worldwide and how it is responsible for energy consumption, corresponding carbon emission, and also different aspects of energy saving methods in the built environment.

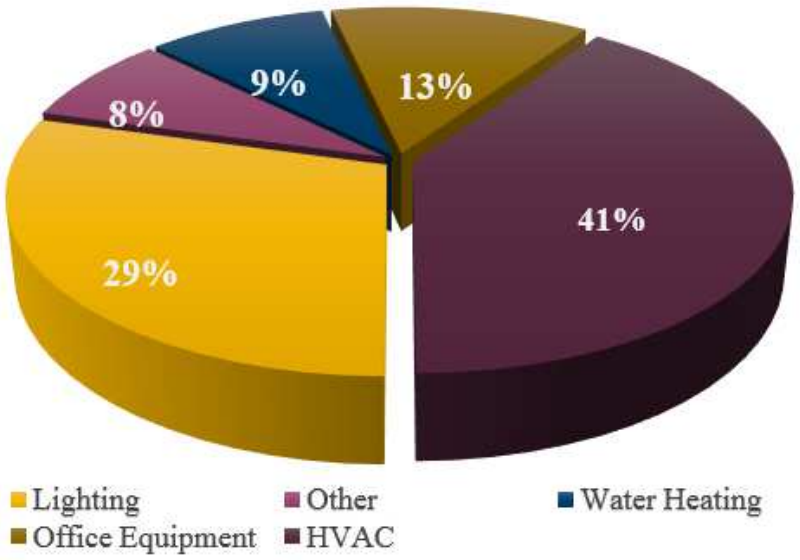


FIGURE 2.1: Overall percentage of energy consumption in commercial buildings by the different building's component.

2.2 Classification of FDD methods

Fault detection and diagnostic processes have evolved, with notable advancements made in terms of data mining and machine learning techniques. Dynamic research and exploration in this field began in the 1980s [18, 19], however from the beginning, practical limitations like scalability and complexity of HVAC systems that have made FDD extremely challenging. According to the literature, FDD can be categorized into several groups depending on their application. In [20], FDD has been classified into three categories after studying around 90 FDD articles prior to 2005, these are: quantitative model based, qualitative model based, and history process based approaches. However, with the on-going advancement of FDD research, other developed methods were proposed later and further categorized FDD into another three groups: analytical based methods, knowledge based methods, and data-driven based methods [14]. One recent review article has been published on the FDD classification in building system by Kim et. al. [10]. Here the authors have reviewed around 200 FDD articles and provide very useful information on new research trends. It has been shown that around 62% of research have been

published on history process, 26% for qualitative model based and 12% on quantitative models. Process history based approaches are derived from measured data obtained from a certain process over a period and fundamental knowledge is derived from it. Further, FDD has been sub-grouped into several categories based on the underlying process to solve it. By thoroughly analysing these recent classification approache here FDD is categorised into three groups: (1) quantitative model based, (2) rule based, (3) process history based. Further, this process history based is subdivided into two categories : (1) knowledge based, and (2) data-driven based (shown in Figure 2.2) depending on the current research requirement.

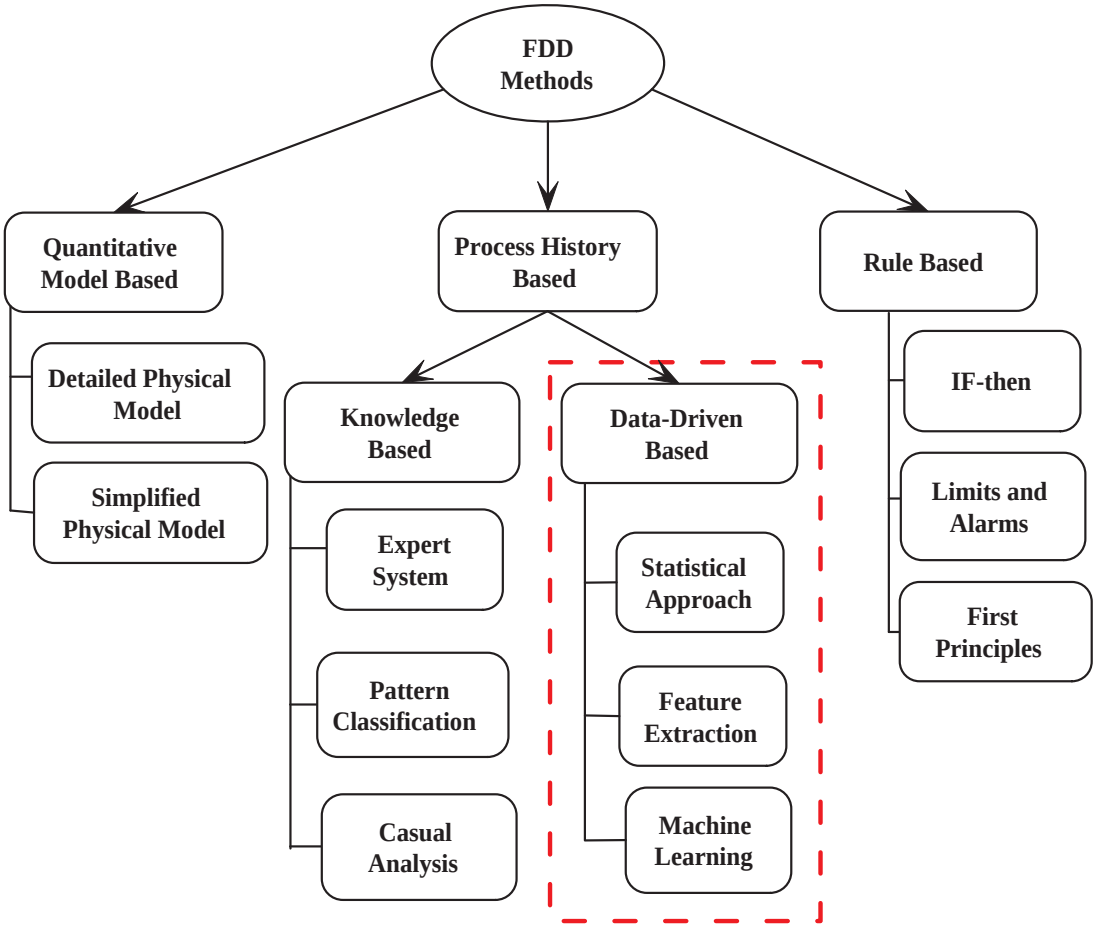


FIGURE 2.2: Classification scheme for fault detection methods.

In Figure 2.2, a data-driven based process is highlighted with red dashed block

and has been employed in this thesis. The requirement of prior knowledge, together with complex modelling processes and heavy computational load, imposes severe constraints on the application of the model-based and rule-based methods. Comparatively, the data-driven methods are affordable since the models normally depend on the available information of the data-patterns. Various data mining techniques have been employed as data-driven methods for FDD studies [21, 22]. These categories are summarized as below.

2.2.1 Quantitative model based methods

Quantitative model-based methods use a mathematical model of a system or plant to achieve analytical redundancy to identify faults and their causes. The theory behind the system is to build each component to simulate the static and dynamic behaviour of the system. These model-based approaches must be validated, and accuracy checked using normal or "non-faulty" and abnormal or "faulty" data before any real application can be made. Model based approaches for building systems can be broadly divided into two types [20, 10]: detailed physical models, and simplified physical models shown in Figure 2.2.

In [23], a detailed physical model was analysed for vapor compression systems. The experimental study is based on actuators such as the variable speed compressor, variable displacement compressor, and electronic expansion valves of vapor compression systems. In this FDD study of vapor compression systems, the use of complicated moving boundary model had been explored. A linear model had been used to discover the sensitivity of the outputs to fault situations of refrigerant leakages, evaporator frosting, and compressor valve leakages. Also, the detailed physical models are obtained using explicit information of the physical connection that controls the behaviour of any system as explained in [24, 25]. Another detailed physical model-based fault diagnosis system is proposed in [26] to control

system behaviour in an online manner. Cooling coil faults and leakage in thermal space were detected, but unknown or new faults have been defined when the difference between the observed output and state of the HVAC system exceeds a chosen threshold. Conversely, simplified physical model-based approaches are simple in terms of mathematical modelling. For modelling, they employ ordinary differential and algebraic equations rather than employing more complex (i.e. partial differential) approaches and simplified physical model is also less computationally heavy. Simplified physical models for temperature control in variable air volume using a feed forward controller was proposed in [27, 28]. Here, a model has been enlarged for real time HVAC applications and shows the ability to control signals according to the system design, but if a large number of issues exist while examining the cause and effect in the system, then performance deteriorated. Several researchers have proposed a non-complex or simplified model which reduces the computational cost of the system.

Shaw et. al. [29], proposed a hybrid model by modifying the simplified physical model for electric load monitoring to reduce system computational cost. In [30] a first-principles based model functionality with expert rule was proposed to handle fault detection in air handling units and results compared with the Gray-box electrical power model. Gray-box is a hybrid approach where constants of an equation of physical models are learned from the data. It was found in this article that the compared models have some limitation in unmodeled phenomena results reducing sensitivity towards robust fault finding. Wu and Sun [31, 32] improved the performance of a real time fault detection and diagnostics system using physics-based auto-regressive moving average model. The parameter estimation and multi-stage regression models have been combined here and were successfully able to predict room temperature in office buildings. Another hybrid model to detect faults for air handling unit was proposed in [33]. Here, the technique was developed by combining statistical residual evaluation and fractal correlation dimension (FCD)

algorithms to detect small bias faults especially under noisy conditions and to identify a small variation of the curve fractal characteristic. This method has been experimented on the simulated data obtained from the TRNSYS simulation platform successfully. Further, support vector regression (SVR) is developed for a model prediction to obtain fault-free references. This technique is limited to detect six fixed biases of the supply air temperature sensor under three different load conditions. A model based multi-input multi-output (MIMO) control of HVAC systems has been presented in [34]. This modelling approach is considered the 'dry effect' due to residual condensation after a dehumidification period which had not been examined earlier was demonstrated by simulation also for a real system and found performance improved over their previous approach in [35].

2.2.2 Rule based methods

Rule-based perspective for fault detection and diagnosis is appropriate when prior expert knowledge is available about the system in large scale buildings [36] to draw a conclusion. The technical information of a system can be applied using specified rules towards making assumptions about the said unit. The rule based approach can be handled in three ways, (a) system with partial knowledge by framing IF-THEN rules, (b) limit and alarms by directly involving measurement from raw sensor data, and (c) first principles that are implemented in a tree structure. Schein et. al., [37] developed a FDD system for operational control with partial or limited knowledge by setting up IF-THEN rules for an air handling unit. Manufacturers have effectively verified this concept for real time building applications [38, 39]. Schein and Bush [40] expanded the concept by proposing hierarchical rules for FDD. The limits and alarms approach is mainly used as rule-based AFDD model to avoid or highlight potential damage in any system [41, 42, 43]. This method, commonly supported by building automation systems, relies on comparing raw outputs that are measured via sensors directly that have expected values, but the

limits for an AFDD system must be defined. Another rule-based solution has been developed via data mining techniques for energy efficiency by the authors in [44]. Here, the authors proposed a set of energy efficiency indicators to detect anomalies. Data mining systems have been developed via knowledge extracted from building sensors to create a set of rules that has been used to make a decision to detect anomalies in smart buildings. In [45], a transient pattern analysis based FDD tool has been proposed for a variable air volume system. This model uses temporary data from five calculated inputs (outdoor and indoor air temperature, damper position, supply air-temperature, pressure). Here an online fault diagnosis tool has been developed. The transient trends in calculated variables while the start-up is compared to the baseline from the regular start-up. Since operating conditions affect the baseline response, it must be normalized before a comparison is made. A first principles-based prototype is generated to detect sensor, economizer damper, and damper actuator faults in an AHU. Then rules which were derived from first-principles are applied to the economizer, cooling coil, and fan duct system. The estimation from the first-principles are used to create (if–then) rules that are assessed by a skilled system using knowledge about the actual process. Generally, a steady-state procedure is employed to filter out transient data, as the first-principles method relies on steady-state operating environments [46, 47].

2.2.3 Process history based methods

A process history based approach is an effective and intelligent ways of analyzing large scale data by identifying recurring patterns and discovering the hidden knowledge from the historical data towards the automatic fault detection and diagnosis purpose. Due to the recent advancements this method has been further classified into two sub categories: knowledge based and data-driven based, described below.

2.2.3.1 Knowledge based methods

Knowledge-based methods in FDD systems consist of observation, analysis and knowledge discovery from the large data available in smart buildings. The knowledge based methods are further subdivided into three different types: casual analysis, pattern classification, and expert systems. Signed Directed Graph (SDG) has been proposed for casual analysis by using the fault symptoms without first principle [48], to minimize engineering effects and to make the fault diagnosis system simple. Capozzoli et. al. [49], implemented artificial neural networks and basic ensembling methods (ANN-BEM) combined with the peak detection method to detect the abnormal condition as outlier conditions of a cluster of buildings. ANNs have characterized a "fault free" hypothesis and a high value represents an anomalous consumption correctly. Detection of evident outliers is performed by coupling the classification and regression tree (CART) algorithm and generalized extreme studentized deviate (GESD) algorithm, which tests for outliers in a univariate data set and approximately follows a normal distribution. Results show the advantage of this data analysis approach in automatic fault detection by minimizing false alarms. Yu et. al. [50] established a four-component framework comprising data mining methods. In addition, they proposed data analysis process, which begins with problem specification to knowledge understanding. Another framework has been developed by the authors [51] which primarily includes four phases: exploration of data, partitioning, discovering the knowledge, and post mining. To resolve the knowledge deficiency of the data, machine learning methods such as the Hidden Markov Model (HMM) [52], Kernel Machines (KM) [53] are applied, where knowledge is automatically extracted from data. Pattern classification algorithms are used to build non-linear correlations between data patterns and fault classes in the absence of clear model structures. Some well-known pattern recognition based methods are Bayes Classifier [54], Artificial Neural Networks (ANN) [53, 55], Support Vector Machine (SVM) Classifier [56], Fuzzy Logic [57] are combined the

rules to identify faults in building. The major aspect of building research and their challenges in terms of discovering the knowledge from the rapidly growing data is studied in [58].

2.2.3.2 Data-driven based methods

The advancement of data-driven approaches for BMS has become increasingly important in the era of big data research. ML techniques have increased in popularity in forecasting the system behaviour due to their advantages in capturing nonlinear and complicated relationships. However, it is a major challenge for building professionals to totally understand the inference mechanism, learn and put trust into such predictions, as the models developed are typically highly complex and have low interpretability. Thus, a method has been proposed to describe and assess data-driven building energy performance models [59]. It can assist building engineers to know the inference mechanism towards prediction. A new metric, i.e., trust, is proposed as an alternative approach other than conventional accuracy metrics to evaluate model performance on actual building operational data. Conversely, data-driven based methods had built relationships between data patterns and faulty classes of a system [60]. These methods extract the key data components and transform the dimension of the whole data. Then, these key components are used instead of the whole dataset for FDD. This approach is fit for modern HVAC systems employed in commercial buildings. Primarily, data-driven has been categoriesed into three approaches: statistical, feature extraction, and machine learning based approaches. Wavelet transformation and short-time Fourier analysis based data-driven FDD procedures are developed [61]. A fusion approach using wavelet transformation and principle component analysis (PCA) is proposed for data-driven fault diagnose of the HVAC air handling unit [62]. It detects faults in large systems using a dimensionality reduction technique that maps

the data to a lower interested space. In practice, this FDD methodology is appropriate for fault detection instead of fault diagnosis. To resolve these constraints, integrated or hybrid procedures are considered for efficient FDD applications in large buildings [63]. Magoules et. al. [21], developed an artificial neural network (ANN) model based on the recursive deterministic perceptron (RDP) ANN to implement FDD at an entire building level. They proposed a new fault diagnostic procedure for detecting and ranking faulty equipment in the order of fault risk. A recent article [64] has developed a method based on stochastic model predictive control (SMPC) and provides a promising solution to complex control problems under uncertain disturbances. In this paper, a SMPC approach is proposed by actively learning a data-driven uncertainty set from the available data by ML methods. Along with the energy optimization method, understanding the occupant's behavior in buildings for improving the energy efficiency has been developed in [65]. A methodology has been developed for classifying occupant activity patterns from plug load sensor data through a case study of an open-office building in San Francisco, USA. A data-driven approach had been combined with physics-based method for forecasting and managing building's cooling and heating system [66]. A statistical approach using generalized Cochran-Orcutt estimation was employed to adjust a linear model for signal data to forecast building's total energy consumption. This forecast technique was then adopted and combined with a model predictive control (MPC) framework to manage heating and cooling set points based on the generated data from the EnergyPlus model. However, this method is limited by a fixed dataset for a few hours of data and needs to adapt appropriately for new and longer data periods [66].

2.3 Conclusion

From the above literature review it can be seen that the BMS domain research for fault detection and diagnosis of HVAC system is extremely challenging and vast. The above mentioned published approaches are mostly restricted to identify specific fault types (e.g., fan failure, stuck valve). Also, most of the research has been performed on model generated data and not experimented on real building data. Most real data research has been found by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). In addition, most of the FDD methods were developed to solve problems related to specific large units such as, chillers, boilers, air handling units (AHUs), and variable air volume (VAV), but none of them focus on small units i.e. fan-coil unit's TU, even though these small units also have huge impact on the building's energy consumption. There is no standardized method present to deal with building energy consumption related to faults, therefore, many energy related problems are left undetected or ignored due to the diverse complex energy systems and uncertain occupant behavioural effects on different building. Additionally, the large number of sensors installed in a building for data gathering keep on processing data in real-time basis, but the location of these sensors are not accurately provided. So manual monitoring and detection of specific fault types are not ideal solutions for performing AFDD efficiently. The building automation system or BMS is trying to solve this core problem and BMS data analysis is novel in that context. Primarily, data understanding is the most important phase to gain insights into the whole system and provide potentially helpful information for further actions. HVAC system consists of several units (AHU, VAV, chiller, boiler, FCU etc.), where each of them are different in nature from each other in terms of physical structure as well as working principles. Thus, controlling and monitoring every unit immediately while the faults occur with a solitary manual method is burdensome for the building experts or the service provider who maintain BMS of a building.

In this thesis, the holistic analysis of a HVAC system highlights a range of common faults in small equipment, i.e. the FCU's terminal unit. It is one of the most challenging units to deal with as these small units are in their thousands in a building and if one of them is not working desirably it could have negative effects on the main plant. Also, they can go unnoticed and bring major impact on energy consumption with such a high number of installations. In fact, characterising terminal unit behaviour using real data has not been given much attention in the current research since they are so plentiful, drawing the attention of this study. Major concern of this investigation is to provide an intelligent and general solution for the building operation and maintenance to deal with any type of faults in this terminal unit.

Chapter 3

System architecture and case studies

"The world is the great gymnasium where we come to make ourselves strong."

-Swami Vivekananda

3.1 Introduction

Large commercial buildings have more complex energy or power demands and could save on losses if the HVAC unit's or any other heavy units are monitored continuously. In a building infrastructure, BMS collects the data from the different physical systems via sensors. BMS provides information related to HVAC units (as well as other units) and the behaviour is mentioned by an expert building engineer or the system provider over time. With human involvement faulty behaviour monitoring of each units can be sometimes ignored or 'best practice not followed' for long time periods that causes significant energy losses. Therefore integration

of resilient intelligence in the built system can help in making an effective building fault detection automation system [67].

In this chapter, the framework for the FDD building system will be explained. Here, the framework is built by a two stage system: (1) the BMS data collection platform previously developed by the company (Demand Logic Ltd.), (2) the key feature extraction of the BMS data and integration with machine learning techniques from real buildings, the PhD work. Combining these two aspects of the BMS study reveals a successful approach in building automation systems, which aims to embed human-like intelligence in this field. The description of the proposed framework for data accumulation is explained, which involves data gathering from the company provided by a dedicated cloud database and the typical big data processing set-up for performing this experiment. Furthermore, the details of two case studies along with a reference floor plan are included later this chapter.

3.2 Proposed framework

The proposed automatic fault detection and diagnosis (AFDD) framework of the entire system is shown in Figure 3.1. This displays step wise development of the system architecture such as data storage, retrieval, analysis, visualization, and support for decision making. Initially, the building data is transferred to a cloud internet through a secured gateway from all of the equipment on the dedicated BMS network. This provides the pivotal information with time data and making it available for pre-processing. The data is first understood i.e., where it comes from, what are the parameters associated, dates and times, etc., and the pre-processes required to incorporate machine learning for different unit's behaviour identification. Then, an automated analysis of exceptional behaviours has been made by the machine learning algorithm so that individual pieces of equipment

can be singled out for attention. These exceptions are raised as notification actions to the building manager to investigate further in details.

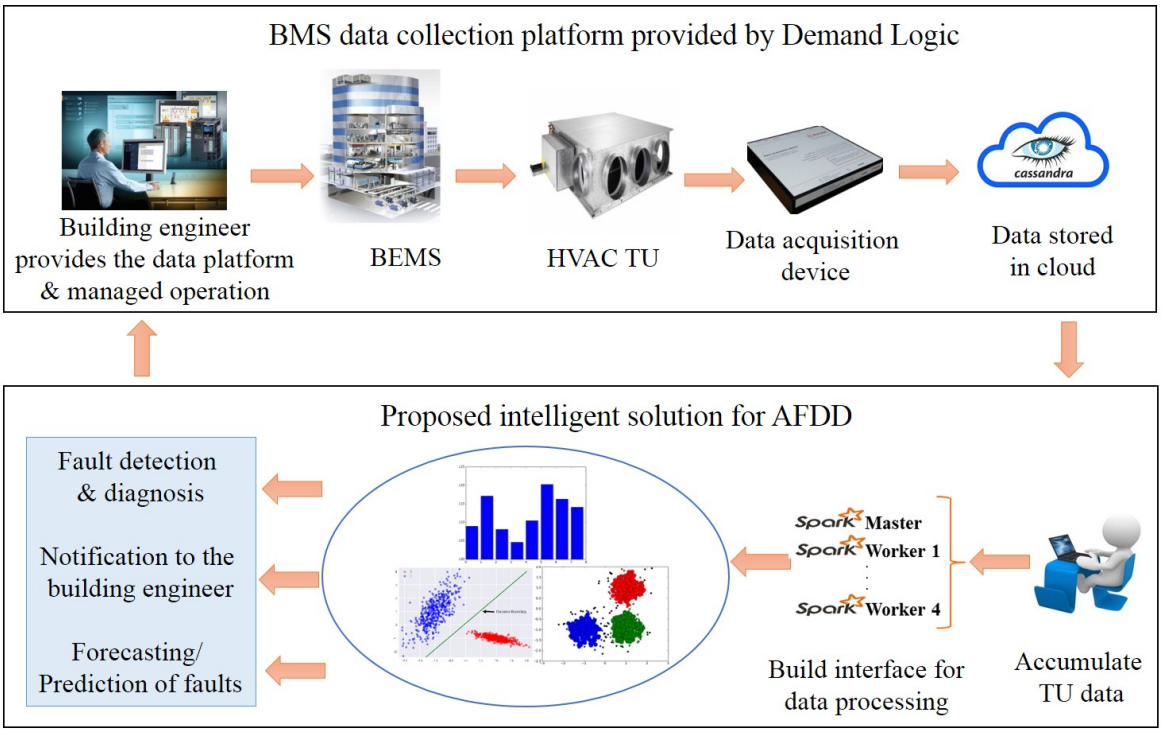


FIGURE 3.1: A framework of the system architecture in this experiment.

3.2.1 Data retrieval process

The data acquisition device (DAD), which acts as a gateway to connect an existing building management system to a secure internet service, is already installed in the buildings under test and is shown in 3.2. The connection can be completed in a single site visit which will take no more than 3 hours in total by the Demand Logic (DL) team. Once this connection is established the remaining set-up is completed remotely again by the DL team. Demand Logic Ltd. is a BMS service provider for several buildings in London and is the partner company for this match funded PhD project. This data gathering process is completed within 24-48 hours and it creates a virtual asset model of all of the equipment on the BMS network and every single data point as BMS data. The BMS data is then extracted via a single

embedded PC connected to the BMS network as a network node (thus the device needs to be given an address on the network, the detail of this depends on the type of BMS). Currently a PC Engine 2D13 ALIX embedded PC is used by the company. The embedded PC is also connected to the internet using either the buildings existing Internet connection or through a mobile network router.

FIGURE 3.2: Data acquisition device (DAD): a company made device for building's data collection.

3.2.1.1 Data collection process

For the data collection an embedded PC (2D13 ALIX) contains implant software that is used as follows:

1. Obtain a map of the BMS network. This includes all the:
    - Local Area Networks (LANs) on a BMS network (a BMS network is more accurately called inter-networks and consists of multiple LANs).
    - Devices on a LAN (a single device may relate to one or many pieces of building services equipment).
    - Data points on a device. These can be binary or analogue control signals, feedback signals or settings.

For each of the LANs, devices and data point the text label and numerical ID is obtained.

2. Polling the values of the data points (typically at intervals of 30 minutes data point type and, by inference, how frequently values are likely to change. For example: the control and feedback signals are polled more frequently while the settings are polled less frequently).

3. Store/buffer the data if the internet connection is lost.

4. Securely send the data to the company connected Cassandra cloud servers.

It has been developed so that a single hardware device can acquire large volumes of data without impeding the operation of the existing BMS. In largest single properties approximately 180,000-200,000 data points are collected every ten minutes. That's seven billion data values a year on the performance of each of these buildings without affecting the automated control of these buildings. This data collection process is carried out as a part of DL's commercial engagement and the data stored in their secured cloud database. These data are then accessed and any subsequent data analysis and machine learning research has taken place in the University lab for this proposed work and is explained in further subsections.

3.2.2 Typical lab set up for parallel processing

The implementation and performance of the proposed method (data analysis and machine learning) rely on the processing architecture in place and is shown in Figure 3.3. The architecture for this work is built for real-time streaming and distributed computing for the large amount of data processing in parallel. To do this, the processing architecture comprises Apache Spark [68], which is an open-source cluster computing framework that enables the scalability and fault tolerance of Map Reduce (programming model for handling huge and unstructured data in

parallel) using resilient distributed data-sets (RDDs). A RDD is a group of objects partitioned through a set of machines that can be reconstructed if partition is lost. Spark in memory runs up to 100 times faster for certain applications by allowing user programs to load data into a clustered memory and querying repeatedly. It is highly suitable for machine learning algorithms [69, 70]. Spark requires a cluster manager and a distributed storage system. Here Spark version 2.0 is used with 5 CPU core of 8GB memory each is set-up by one master and four workers and the connection of the distributed storage with the Apache Cassandra interface. The experiments have been performed using 64-bit Java version 8. By using this set-up the computational time become four times faster than normal processing.

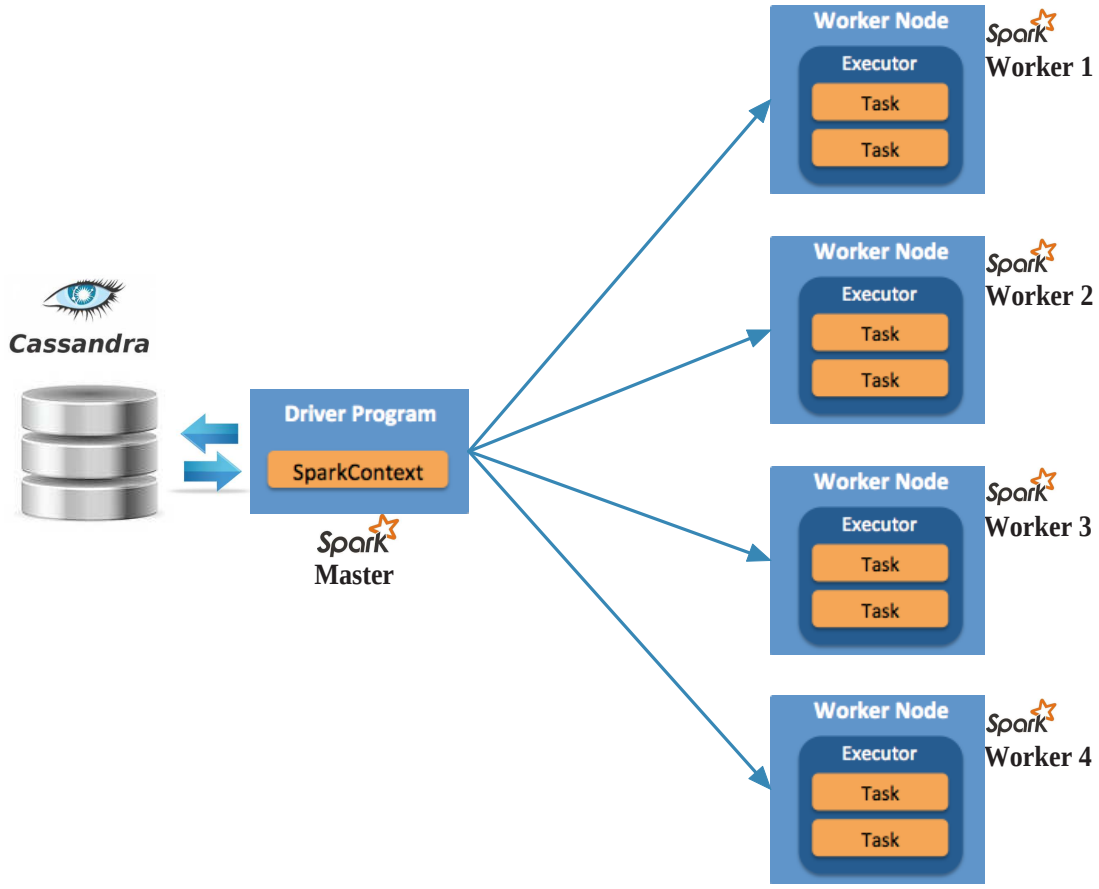


FIGURE 3.3: Set-up of parallel processing architecture.

3.2.2.1 Data characteristics

The BMS data used follows the characteristics of "4*V*" [71], i.e., volume, variety, velocity, and value which reflects the big data characteristics in nature. All these pre-defined characteristics which are related to these BMS data explained below:

Data Volume: The data is continuously generated and increasing in volume. In this experiment, more than 50 million data points are considered.

Data Variety: The data type is not traditionally structured in nature, but on standard wise mixed of structured, semi- structured, and unstructured formats. There are six types of data are included such as, time stamp, voltage, temperature, binary signal on and off. This complete set of different attributes make a composite structure which needs a complex processing logic to handle the aggregation.

Data Velocity: The speed of the data flow is very fast. The massive and continuous data is stored in a Cassandra cloud from the source device (here, DAD). Approximately one million data points are gathered every day for a single building, which help in making valuable decisions that provide strategic advantages after handling the data velocity.

Data Value: These data are as valuable as the building outcomes. It provides awareness of the behaviour of TUs, which increase the potential to implement and improve fault analysis, decision making capabilities and measure faulty equipment for diagnosis and energy saving possibilities. Therefore, this determine the ultimate value of the collected data.

As mentioned in Chapter 1, six different parameters (set point, dead band, control temperature, enabled, heating effort, cooling effort) are specifically selected for this work. Based on these parameters the demand and the control strategies are

varied and deal with the real time TU issues as mentioned in Section 1.3. The buildings under test raw data comprises information such as, time stamp, voltage, temperature, valve opening percentage, etc. To process these real data there are several problems as, would be expected in real environments, different sampling rates for different attributes, variable arrival time, missing values for intermediate time stamps, etc. Therefore, the TUs with the missing data are filtered and linearly interpolated to re-sample data at regular ten minute time intervals throughout this investigation process. This approach can be used in other settings.

3.3 Case studies

The case study is based on two large commercial buildings in central London, U.K. The investigation began with TU data collection for fan coil units (FCU) from these real buildings over a specific period from July 2015 to July 2018. Initial experiments started with one building's historic data from the year of 2015 and the entire experimental study has been based on these data by visualizing the characteristics and discovering patterns. Data from 2016 to 2018 period has been employed to perform further experiments such as, seasonal analysis and fault prediction. A floor plan for all FCUs distribution across the first floor is displays in Figure 3.4 to show the actual floor-wise FCUs positions that are pointed by a blue box in the figure.

3.3.1 Case study-1

The building for case study-1 was built in 1960 (London), which was renovated later in 2009. It covers 149,000 sq. ft. for offices and 8,000 sq. ft. for retail space. This building has seventeen (17) floors with seven hundred and thirty one (731) terminal units distributed across the different floors. The floor plan

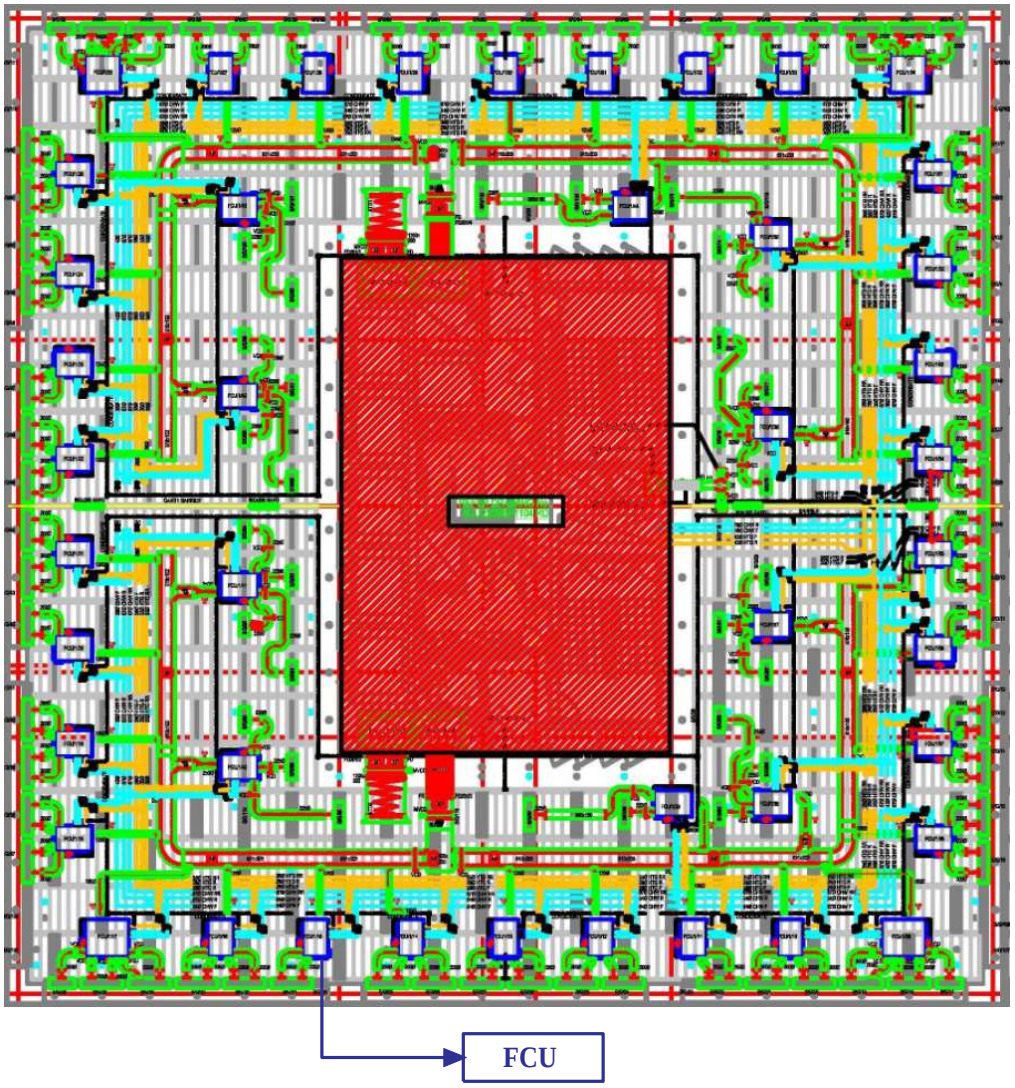


FIGURE 3.4: Floor plan for all FCU's distribution across first floor- for case study-1.

drawing of this building is observed closely to understand relationships between HVAC TUs, rooms, spaces, and other physical features at one level of a structure. There are 44 FCUs fitted in a floor area. In this building out of 731, 723 TUs are operating normally and most of them are well maintained thus it is considered as a well operating building. Therefore, this research has began by investigating the behaviour of the TU of this building. Initial testing started from the year of 2015's data after 17th July and continued till July of the year of 2018's data. The 2015's data has been investigated thoroughly and used for the training and testing purpose. Thereafter, all the 3 year data have been used to classify TU behaviour

and to perform real time prediction. In addition to this case study, to investigate the robustness of the AFDD system, another building has been taken into account and is described subsequently.

3.3.2 Case study-2

This building comprises of seven floors and all together five hundred and sixteen (516) TUs across the different floors. Each floor plan comprises HVAC TUs, rooms, spaces, and other physical features. This building's units are comparatively bad behaving to case study-1. Approximately, 490 TUs are operating properly among all TUs. Thus, this building is chosen as the second case study to identify different TU behavioural patterns. The TUs of this building have been used for automatic fault detection and diagnosis and seasonal analysis for semi-supervised learning. This test has been performed to validate the robustness of the proposed methods for different buildings. The experiment, for this case study began from 1st of January 2018 until 31st July 2018. The details of the chosen data for the specific experiment is tabulated in Table 3.1

TABLE 3.1: Building's data details for both the case studies

Buildings	Total TUs	Operating TUs	Period from	Period to	Experiments
Case study-1	731	723	17/07/2015	31/07/2018	Unsupervised, Supervised, and Semi-supervised
Case study-2	516	490	01/01/2018	31/07/2018	Semi-supervised

3.4 Conclusion

The chapter provides a summary and appreciation of the system architecture and data collection provision. The architecture provides a stable baseline into which

incoming data can be pre-processed, managed and the effectiveness of the machine learning methods monitored. The details of the data gathering have been provided by the company using DAD through a specific gateway is elaborated. The characteristics of these collected data has been explained. Although the work is conducted on two specific case studies over a given time, initial experiments began by investigating the daily data of 17^{th} July 2015 and then progresses to week, month, and yearly data over a period of three years. From this date the data has been gathered, analysed, tested and considered as historic/old data to learn the pattern via ML models for fault identification in this thesis. Thereafter, details of the proposed feature extraction of these collected HVAC terminal unit data is presented in the next chapter.

Chapter 4

Proposed feature extraction

> "Take up one idea, make that one idea your life, think of it, dream of it, let the brain, muscles, nerves, every part of your body be full of that idea, and just leave every other idea alone. This is the way to success."
>
> *-Swami Vivekananda*

4.1 Introduction

Building data - the key challenges associated with a building's 'big data' is how to condense the information contained therein to be suitable for modelling without incurring notable information loss. In the real world, data is coming from heterogeneous sources and are subjective, hence not fit for purpose, thus interpretation of data knowledge is difficult. To deal with the data deluge and make the best possible use of the available data, building researchers look to the ever increasing data

volumes on trends thus rolling micro-scale management of the data. Improvement and informed strategies require multi-disciplinary approaches in order to use the best of building research fields to implement real-world problems.

In this chapter, the current explosion of data volume, recording and attributing, has initiated the expansion of many big data platforms along with parallel data analytic procedures and machine intelligence techniques. The quality of the raw data is not always appropriate in form due to the lack of collaboration, standards, etc. (e.g. data are not always in the correct format, missing values, and so on). This happens due to the manual installation process in buildings, which can cause significant complication when dealing with fault identification of associate units. So, the data produced in whatever form is pivotal to the whole system. Conversely, huge datasets might produce worsening performance in data analytic understanding. Thus, this has pushed towards data dimensionality reduction procedures, and unfortunately not always with better results [72]. The presence of time series terminal unit (TU) data increases computational costs due to the explosion in input dimensionality (potentially hundreds or thousands of samples instead of a single value) and makes it difficult to interpret and exploit knowledge in their current format. Thus, in this work a milestone was to reduce the dimension of the datasets to represent the data. For this purpose, a novel feature extraction method is emphasised [73] to transform the data into a lower dimensional space in a unique way. The whole process is developed by sets of application-dependent features; such a process is also known as feature engineering. This process is mandatory to minimize computational overhead of the system and is the intermediate step in this thesis, to create an effective machine learning model for the proposed automatic fault detection and diagnosis (AFDD) study explained below.

4.2 Proposed feature extraction method

An 'intelligent' feature extraction method [73] is proposed to deal with the available high dimensional terminal unit (TU) data. It is an 'attribute reduction process', which projects datasets of higher dimension into a smaller number of dimensions intended to be informative, non-redundant, facilitating subsequent learning and for improved on automation. This data-driven feature extraction method is accomplished based on the six TU parameters: control temperature, set point, deadband, heating effort, cooling effort, and enable signal have previously been described in Section 1.3.1. These time series data is collected at 10 minutes interval throughout a day by a secured gateway (described in 3.2.1). The proposed feature extraction method has been employed by considering three different events from the area under the control temperature and the corresponding power curves of a single TU data. The events are divided into three different stages based on their flow during a whole day (24hrs) described as follows:

1. Event discovery stage
2. Event area calculation stage
3. Event aggregation stage

4.2.1 Event discovery stage

Event discovery is inspired by the proportional integral derivative (PID) controller response curve [74] as shown in Figure 4.1. PID is a combination of proportional, integral and derivative control, with the combined operation of these three controllers provides control strategy for the output of the process plant. A PID controller manipulates the process variables such as pressure, speed, temperature,

flow, etc. and is a control loop feedback mechanism to accurately control the process, remove oscillations and increase efficiency. It is extensively used in industrial control systems to regulate the process variables. Proportional or P-controller gives a controller output that is proportional to current error. Integral or I-control integrates the error over a period until the error reaches to null. With derivative or D-control the error changes rate with respect to time and can react once the set-point has changed.

A typical step response curve following a controller responds to a set point alteration is shown in 4.1. The curve rises within a given period from 10% to 90% of the final steady state value in a time known as the rise time. The percent Overshoot is the amount that the process variable overshoots the final value, expressed as a fraction of the final value. The settling time is the time required to settle within a certain percentage (commonly 5%) of the final value. The steady-state error is the ultimate difference between the process variable and set point. Dead time is a delay between when a process variable changes, and when that change can be observed. For instance, if a temperature sensor is placed far away from a cold water fluid inlet valve, it will not measure a change in temperature immediately if the valve is opened or closed. Dead time can also be caused by a system or an output actuator that is slow to respond to the control command, for instance, a valve that is slow to open or close, can cause dead time. A common source of dead time in HVAC systems is the delay caused by the fluid flow through pipes. PID controllers minimize the error over time by adjusting a control variable and hence produces output that is applicable to plant control devices.

Based on the response curve shown in Figure 4.1, the pre-processed data streams retrieved from an individual TU are divided into different events in an event discovery stage. In this stage, the events are selected from control temperature and corresponding power effort data streams. Inside a building, when the heating and

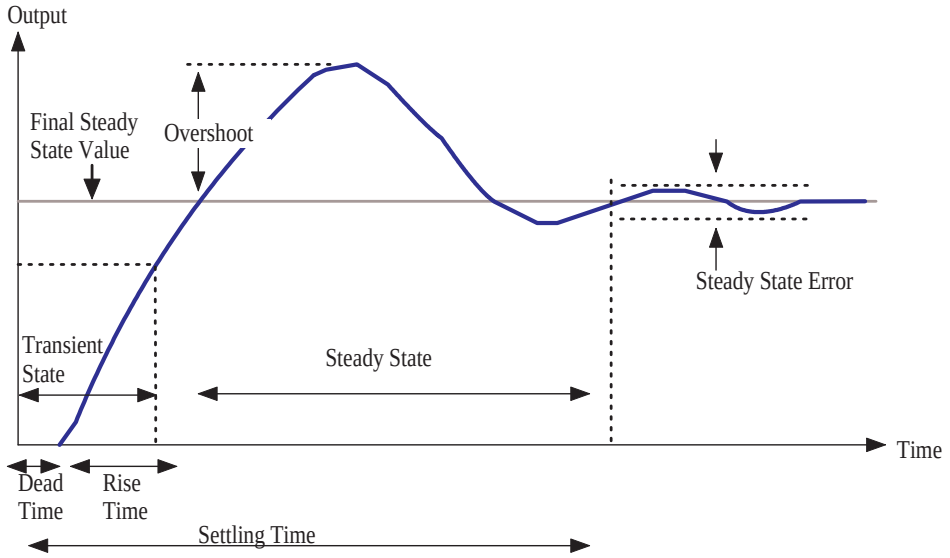


FIGURE 4.1: PID controller response curve.

cooling units are started during normal operational hours, the temperature begins to change depending upon the environmental request. Therefore, two types of events, heating event and cooling event are selected depending on the corresponding power demand at that time instance. Based on the control temperature variations with respect to the set point value, the data stream is divided further into four different stages: event start, response delay, goal achieved, and event end, which happen in a single day. A single day TU's event discovery stages are divided into four stages are described below and displays in Figure 4.2.

1. Event start (ES) : An event start is assumed to happen when the BMS is first switched on a particular day (enabled signal gets switched on) and the time instant when the temperature starts to change.

2. Response delay (RD) : It has been mentioned previously that due to the process variable delay during the dead time, the temperature starts to respond only after a certain delay from the previous point when the BMS is switched on and this event is termed as response delay. This is essentially the time

spent by the TU during dead time as shown in Figure 4.1. The delay is anticipated when the temperature starts to respond only after a certain delay from the previous point after enabling BMS.

3. Goal achieved (GA) : A goal achieved event is assumed to happen when the control temperature reaches the desired set point. GA can be considered as the time instant when the process variable reaches the steady state, or final value. This is essentially the time spent by the TU during rise time as shown in Figure 4.1.

4. Event end (EE) : Once the control temperature reaches the set point, it may either continue to be within the dead band till it exceeds the dead band, and an event end is supposed to happen at that time instant. This is essentially the time spent by the TU in the steady state and there could be a percentage overshoot above the final set point value as shown in Figure 4.1.

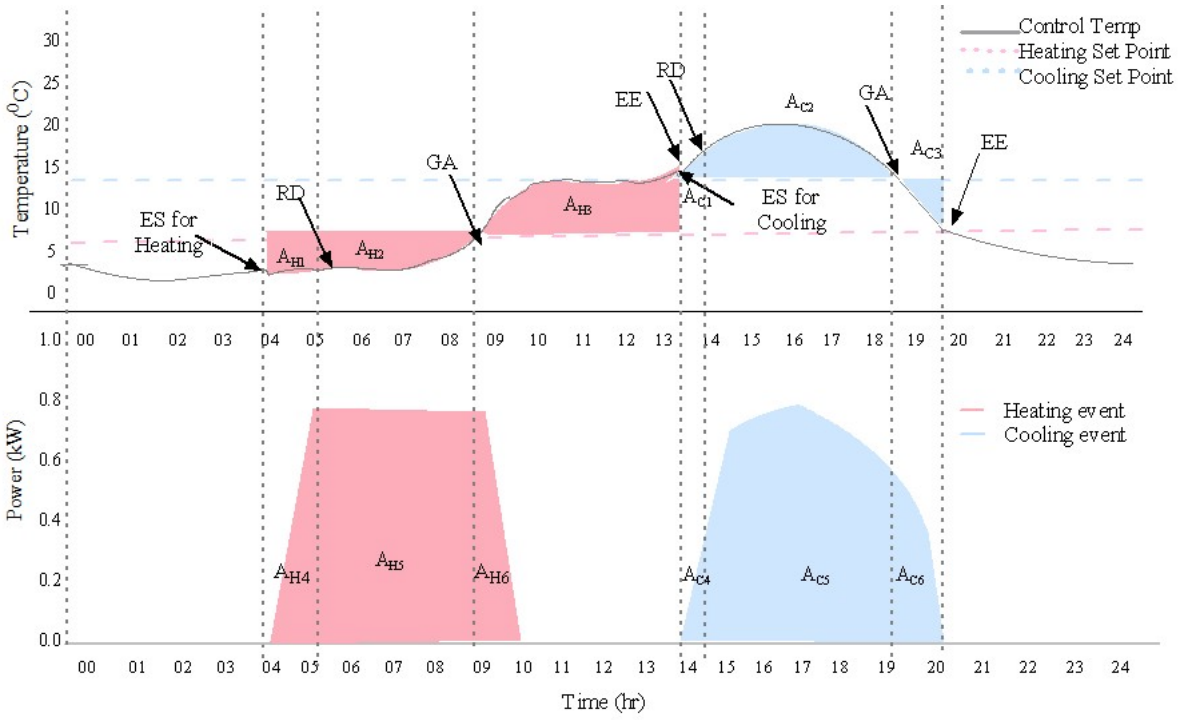


FIGURE 4.2: Event discovery process of one day TU.

In Figure 4.2, pink colour indicates the heating power demand and blue colour indicate cooling power demand for both the control temperature (in degree centigrade) and associated power demands (in kilowatt) of a whole day (for 24 hrs). The different events are also indicated by arrow in the control temperature graph and for power the dotted lines are drawn to show different events. In this stage four events have been discovered to represent the temperature as well as power variation throughout the day for both heating and cooling activity.

4.2.2 Event area calculation stage

After discovering the appropriate heating and cooling events, the estimated area under the temperature and power curve for each event has been calculated. Subject to the event type area calculations are carried out respectively for both heating (H) and cooling (C) events. In effect, six areas (three from temperature and three from power curve) for each heating event and similarly, six areas for each cooling event are calculated. Altogether twelve different areas are derived from all the daily TU data.

The area (A_E) beneath the curve $f(x)$ at every time interval Δx is given as:

$$A_E = \sum_{i=0}^{n} f(x_i)\Delta x \tag{4.1}$$

In a heating event, the area calculations for temperature are indicated by A_{H_1} *to* A_{H_3} and for power indicated by A_{H_4} *to* A_{H_6}. Likewise for a cooling event, the area calculations for temperature are indicated by A_{C_1} *to* A_{C_3} and for power are indicated by A_{C_4} *to* A_{C_6}. After these areas have been computed, they are normalized to obtain the final feature values as F_{H_1} *to* F_{H_6} and F_{C_1} *to* F_{C_6} as denoted by Equations (4.2) to (4.5).

The area calculations for a heating events are shown as:

$$F_{H_1} = \frac{A_{H_1}}{T_{H_1}}, \quad F_{H_2} = \frac{A_{H_2}}{T_{H_1}}, \quad F_{H_3} = \frac{A_{H_3}}{T_{H_2}}$$
$$where, \; T_{H_1} = max(A_{H_1} + A_{H_2}) \tag{4.2}$$
$$and, T_{H_2} = max(A_{H_3})$$

$$F_{H_4} = \frac{A_{H_4}}{P_{H_1}}, \quad F_{H_5} = \frac{A_{H_5}}{P_{H_1}}, \quad F_{H_6} = \frac{A_{H_6}}{P_{H_2}}$$
$$where, \; P_{H_1} = max(A_{H_4} + A_{H_5}) \tag{4.3}$$
$$and, P_{H_2} = max(A_{H_6})$$

The area calculations for a cooling events are shown as:

$$F_{C_1} = \frac{A_{C_1}}{T_{C_1}}, \quad F_{C_2} = \frac{A_{C_2}}{T_{C_1}}, \quad F_{C_3} = \frac{A_{C_3}}{T_{C_2}}$$
$$where, \; T_{C_1} = max(A_{C_1} + A_{C_2}) \tag{4.4}$$
$$and, T_{C_2} = max(A_{C_3})$$

$$F_{C_4} = \frac{A_{C_4}}{P_{C_1}}, \quad F_{C_5} = \frac{A_{C_5}}{P_{C_1}}, \quad F_{C_6} = \frac{A_{C_6}}{P_{C_2}}$$
$$where, \; P_{C_1} = max(A_{C_4} + A_{C_5}) \tag{4.5}$$
$$and, P_{C_2} = max(A_{C_6})$$

4.2.3 Event aggregation stage

Multiple heating and cooling events can occur during a single day therefore, all the events of a given type must be aggregated to represent the averaged values.

Thus, the next step in the feature extraction process is event aggregation. The final aggregated features can be shown as:

$$F_{H_k} = \frac{1}{n}\sum_{i=1}^{n}(F_{H_{k_i}}) \tag{4.6}$$

$$F_{C_k} = \frac{1}{n}\sum_{i=1}^{n}(F_{C_{k_i}}) \tag{4.7}$$

In Equations (4.6) and (4.7), k denotes the number of events and n denotes total number of occurrences for each event in both the heating and cooling event type. Thus, a single day TU data can be represented using twelve features for both the heating and cooling events. This stage will create the final feature vector for machine learning application to identify different fault from the TU data.

4.3 Event-wise feature mapping

The proposed feature extraction algorithm (presented in Equations (4.2)-(4.5)) aims to reduce, measure, and build derived values from these initial datasets and is intended to be informative and non-redundant, aiming to facilitate subsequent learning and generalization steps. The raw TU data for the control temperature and power altogether contains two hundred eighty eight values to represent a single TU, is now transformed into twelve-dimensional (FC_1 to FC_6 and FH_1 to FH_6) value to represents the same TU in 24 hours. The visualization of these multivariate features in a form of 2-dimensional chart for a single TU are illustrated by radar chart or graph in Figure 4.3.

This radar graph shown in Figure 4.3 is a circular display with twelve different quantitative axes. Each axis represents a feature which signifies the fluctuation of

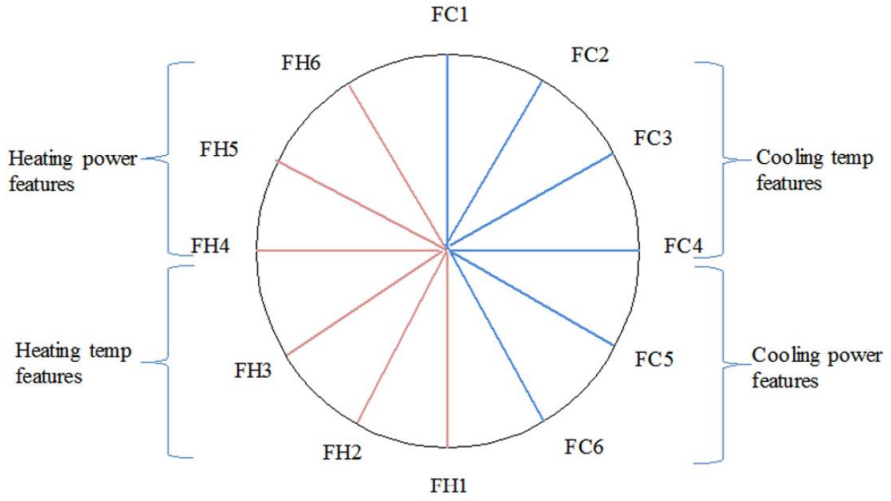


FIGURE 4.3: Description of radar graph.

the temperature and power. The centre of the radar is zero valued and the edge point is with maximum value of one as all the feature value has been normalized between 0-1. The feature values are plotted along each axis which forms a unique shape to represent a TU performance on a daily basis.

The proposed feature extraction algorithm (described in Section 4.2) generates a feature vector of 12 numeric values which exhibit the cooling and heating characteristics of a TU. Thus, the radar is indicated with two colours defines two different parts, first six blue coloured axes (FC_1 to FC_6) are represent cooling events and next six axes (FH_1 to FH_6) are represent heating events including the temperature and corresponding power variation. The first axes of each type of event, FC_1 and FH_1 indicate the temperature state from the event start (ES) to response delay (RD) are calculation. FC_2 and FH_2 indicate the time of response through the calculated area from response delay (RD) to goal achieve (GA). FC_3 and FH_3 describe the event end after control temperature reaches the specific set-point by calculating the area within the goal achieve (GA) to event end (EE) stages. Similarly, FC_4 to FC_6 and FH_4 to FH_6 represent the power demand for corresponding temperature change by the TU data.

Hence, it is to be further observed that greater areas under GA to EE area (A_{H_3}

and A_{C_3}) represent a "well behaved" TU, because this implies that the TU achieves the set point (goal) and spends more time within the dead band (desired range between heating and cooling set point). Whereas the area under ES to RD (A_{H_1} and A_{C_1}) and RD to GA (A_{H_2} and A_{C_2}) is larger denotes that the temperature is not within the dead band or takes a long time to reach the set point represent a "badly behaved" TU. Also, the more area for power effort indicates the more power consumption which signifies a "badly behaved" TU as well as shown in Figure 4.2.

4.4 Conclusion

This chapter has described the proposed feature extraction process to represent the huge volume of building data to discover the valuable information/knowledge within it without any distinguished information loss. This is an intermediate and pivotal step in the proposed data driven automatic fault detection and diagnosis research. Machine learning techniques often suffer from the curse of dimensionality problem; thus, it is difficult for end users to find appropriate patterns that can define system behaviour. This feature extraction/feature engineering method provided here gives an intelligent method to represents terminal units and their correlation with associated temperature and power demands.

The novel feature extraction method is supported by a proportional integral derivative (PID) controller response curve to discover events from the terminal unit (TU) data on a day. Twelve features have been extracted from a daily TU data. Among them, six features represent cooling and the remaining six features represent heating events. The detailed theory has been presented to provide an understanding of this feature extraction and how these new features have been mapped with TU behaviour through radar representation. These new TU features are further employed for clustering and classification purpose to identify distinct faulty and

non-faulty TU patterns for a building at a given time. The details of the implemented methods and experimental results are explained in the following chapters.

Chapter 5

Unsupervised learning

"You are the creator of your own destiny"

-Swami Vivekananda

5.1 Introduction

The scheduling and maintenance of HVAC units are generally controlled by the building engineer and/or building's service provider, thus there is always a lack due to manual processing while monitoring the unit's behaviour. Such limitations could be handled efficiently and quickly by adopting appropriate machine learning (ML) techniques. A HVAC Unit's fault can occur any time of the day and in any form during their operation without any prior information. Also, the causes and sources of the occurring faults are unknown, making fault identification more difficult. Unsupervised machine learning is suitable in this type of building's fault identification scenario. However, prior to that the data must be correctly pre-processed in an appropriate format to apply ML techniques. Hence, the new derived features (described in the previous chapter) from the HVAC terminal unit

data are fed into the ML model to perform fault detection. As the labels/ prior knowledge of the TU's characteristics are unavailable, unsupervised learning has been employed to mine the data patterns automatically and to understand the distinct TU behaviours.

This chapter describes the unsupervised learning or clustering techniques using the new extracted featured data to understand distinct TU data patterns. The most appropriate clustering method has been investigated and employed to develop an automated fault detection and diagnosis tool for use with HVAC terminal unit. An experimental analysis has been made on the outcomes of each clusters in terms of faulty and non-faulty characterization. A comparative analysis and statistical validation have been performed to check the robustness of the employed methods towards the faulty and non-faulty TU behaviour identification.

5.2 Different types of clustering approaches

Due to the unavailability of prior information about the TU data involved in this investigation, an unsupervised learning methodology is employed to discover a set of TU behaviours. This unsupervised learning or clustering algorithm learns to interpret data patterns from the input data without any label information [1]. Clustering similar groups types of data into the same cluster and dissimilar data into another without knowing any physical information about the data.

Several algorithms have been proposed and developed to perform unsupervised learning tasks and are primarily categorized into four different types, shown as a taxonomy in Figure 5.1. These are: partitioning based, hierarchical based, grid based, and density based [75]. Partitioning clustering divides n number of objects in a database into k partitions, where each partition is represented as a cluster [76]. K-means, K-medoid, fuzzy C-means, etc. algorithms fall under this partition based

category [76]. Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom. Two types of approaches:divisive and agglomerative are such examples [76, 77]. In grid-based clustering, the data space is quantized into finite number of cells which form the grid structure and perform clustering on the grids. The density based clustering locates regions (neighborhoods) of high density that are separated from one another by regions of low density [77]. These stated clustering algorithms are suitable for specific tasks which depends on the dataset. Also, different ordering can create totally different clustering results, as to the cluster number generation and also within the clusters itself. However the correctness of the clustering algorithms is effectively relies on the data distribution.

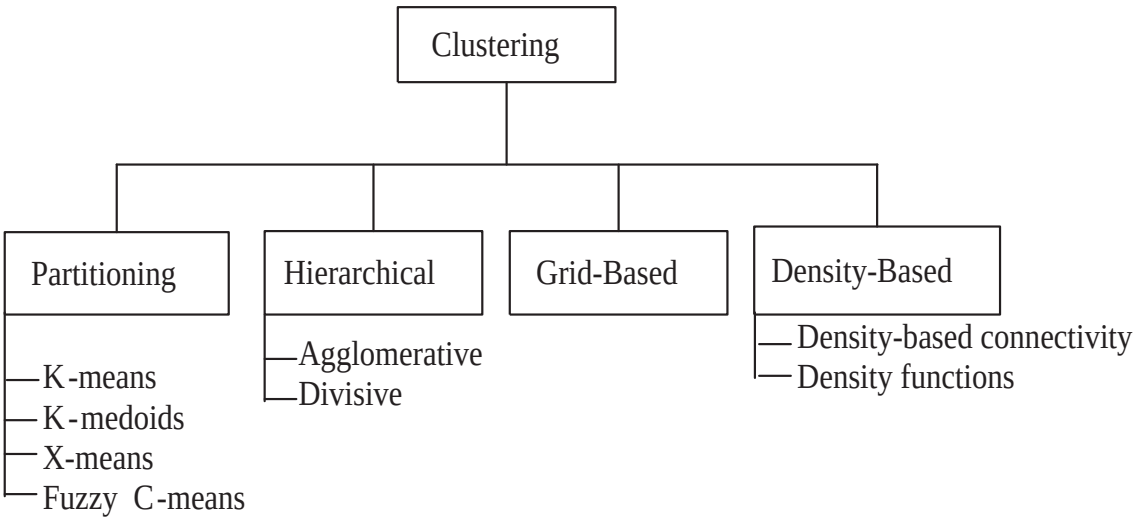


FIGURE 5.1: Basic categorization of unsupervised learning techniques [1].

Here, a clustering algorithm is employed to characterise and categorise distinct TU behavioural patterns. The number of clusters are not known a priori in this experiment. In fact, new groups are forming as the algorithm evolves, thus making it difficult to discover appropriate clusters that represent the whole TU dataset. Hence, this investigation began by exploring X-means clustering algorithm as it is comprises two steps: Bayesian information criterion and k-means algorithm. This helps to first identify the optimal number of cluster selection for the dataset and

then cluster them based on the proximity measure by k-means. Further, results are compared with hierarchical and k-medoid algorithms and validated through the statistical model discussed in the result analysis section.

5.3 Proposed clustering methods

The proposed AFDD method has been performed in four consecutive stages with a flow chart presented in Figure 5.2. Initial work began by collecting the historical TU data from the company provided cloud database (described in Section 3.2.1). Thereafter, a feature extraction method has been developed and performed on the raw TU data (described in Section 4.2). Subsequently, extracted new featured data have been used to perform clustering for faulty TU and non-faulty TU behaviour identification [78].

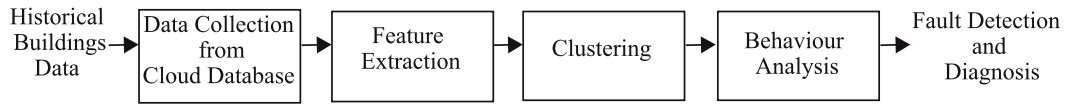


FIGURE 5.2: Steps involved in proposed unsupervised learning techniques.

5.3.1 X-means clustering

Extended K-Means (X-means) [79] clustering is employed on the extracted TU features (obtained from the Equations (4.6) and (4.7)). This clustering is used to avoid the limitation of conventional K-means clustering, which aims to automatically determine the number of clusters based on Bayesian information criterion (BIC) scores. Initially, the whole dataset is considered as a single cluster and the BIC score is calculated using Equation (5.2). Based on the BIC score, the optimal cluster number for that particular data set is chosen for clustering. The

conventional K-means algorithm is then performed by varying cluster seed value. The following X-means steps are as follows:

Improve-params (k-means)- K-means is evaluated by the seed value of the cluster number. The objective function used for this work is shown as,

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||X_i^{(j)} - \mu_j||^2 \tag{5.1}$$

Where, $||X_i^{(j)} - \mu_j||^2$ is the Euclidean distance measure between a data point $X_i^{(j)}$ and the cluster centre μ_j. It is an indicator of the distance of the n number of data points from their respective cluster centres. The main idea is to define K centroids, one for each cluster. Each data point is assigned to the group that has the closest centroid. After all the data points are assigned, the positions of the K centroids are recalculated. The above steps are reiterated until the centroids no longer move.

Improve-structure (BIC)- In this stage, the BIC score is calculated based the clustering outcomes. Maximum likelihood (the parameter estimation in a given observation) of the current clusters is used to determine the BIC score. Centroids are further broken down based on the BIC score in order to discover improved fitting of the data. The BIC is computed here using,

$$\begin{aligned} BIC_{score} &= -2\log(L) + K\log(n) \\ where, L &= P(x \mid \theta, M) \end{aligned} \tag{5.2}$$

Where, L is the maximum value of the likelihood function of the model M. The other parameters, x, θ, n, and K denote the observed data, the parameter of

the model, total number of data points and the number of free parameter to be estimated respectively.

K>Kmax- Once the centroid (k) is determined and K-means is performed, K_{max} is then selected, and all centroids are tested. As the lower BIC score is always preferred for better fitness of the data, the BIC score is then compared between K and K_{max}. If the current model has a better score, then the split is considered the best strategy for clustering.

5.3.2 Hierarchical clustering

Hierarchical clustering creates a hierarchy of clusters. It clusters the dataset by measuring linkage criteria between the sets or the groups of observations. It is a function to measure the pairwise distances between the observations in each set [80, 81]. There are mainly three types of linkage used: complete linkage define as,

$$maxd(a,b) : a \in A, b \in B \tag{5.3}$$

The single linkage define as,

$$mind(a,b) : a \in A, b \in B \tag{5.4}$$

The average linkage define as,

$$\frac{1}{|A|.|B|} \sum_{a \in A} \sum_{b \in B} d(a,b) \tag{5.5}$$

Where, a and b is the object which is belongs to the set of A and B. It generally falls into two types: agglomerative and divisive.

Agglomerative- It is a bottom up process where each object in the dataset is assumed as a separate cluster and subsequently merged based on the similarity between them. Likewise it moves up the hierarchy until it forms distinct clusters.

Divisive- It is a top down process where all the objects in the dataset are considered as one cluster and splits are performed based on the dissimilarity found between the each objects. This process done recursively and move down the hierarchy. Hierarchical clustering represented by a dendrogram.

5.3.3 K-medoid clustering

Similar to k-means, k-medoids is a partitioning based clustering algorithm. It is also known as Partitioning Around Medoids (PAM) [82]. In summary, it clusters the dataset of n objects into k number of clusters by assuming k is known a priori from the BIC. It partitions the data set based on the distance based function. K-means attempts to minimize the total squared error, while k-medoids minimizes the sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster. This method is generally more robust to noise and outliers as compared to k-means because it minimizes a sum of general pairwise dissimilarities instead of a sum of squared Euclidean distances and defined as,

$$||a - b||_2 = \sqrt{\sum_i (a_i - b_i)^2} \tag{5.6}$$

Where, a and b is the two different data point in the data set and i is the total no of data.

5.3.4 Clustering internal evaluation

Clustering is an unsupervised technique to group data sets into some specific meaningful areas but the knowledge is not being provided externally to evaluate the grouping technique. For example, a buildings TU data have been divided into two groups instead of knowing which groups refers to what characteristics. Therefore, to know how well the data is being separated, the compactness within each group must be measured. If the compactness value is high then it is assumed that the clustering is successful and the data points belonging to each cluster are similar in nature. If the compactness value is low then the clustering needs further investigation. Thus, in statistic a validation method is present to evaluate the effectiveness of an unsupervised learning technique where the external known class labels are not available. This is called internal validation criteria. This process measures the degree of intra-cluster cohesion and inter-cluster separation. In this thesis, the evaluation of the clustering technique have been performed by using two criterion: Davies-Bouldin (DB) and Silhouette (SI) .

- Davies-Bouldin index- The Davies-Bouldin index is defined as

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{j\neq i}\frac{\delta_i + \delta_j}{\Delta_{i,j}} \tag{5.7}$$

  Where, $\Delta_{i,j}$ is the clusters distance ratio for the i^{th} and j^{th} data point within to between cluster. ∂_i and ∂_j is the average distance between each point in the cluster from the centroid of that i^{th} cluster and the average distance between each point in the j^{th} cluster and the centroid of the j^{th} cluster [83].

- Silhouette index- The silhouette indexing [84] is a measure of the similarity of each point with other points in its own cluster, when compared to points in other clusters. The silhouette value is defined as

$$SI = \frac{1}{nk} \sum_{i \in k} \frac{b_i - a_i}{max_{(a_i, b_i)}} \tag{5.8}$$

Here a_i is the average distance from the ith point to the other points in the same cluster, and b_i is the minimum average distance from the ith point to points in a different cluster.

The silhouette value ranges from -1 to +1. A high value indicates that i is well-matched to its own cluster. The clustering solution is considered appropriate if most points have a high silhouette value. Whereas, a low DB value consider the clustering solution is appropriate.

5.4 Clustering result analysis

After employing the above clustering methods, it is experimentally found that due to data uncertainty, the hierarchical clustering and K-medoid are unable to represent the whole TU data into meaningful groups. Hence, BIC is being performed as a part of X-means clustering on the TU featured data and found the optimal score is achieved in six numbers to represent these TU data. Thus, all three clustering methods have been performed by considering six clusters as standard to represent these buildings data from the chosen case study (described in Section 3.3). Thus, the experimental analysis has been performed by considering the X-means as a baseline method and further two methods have been compared and validated to support the positiveness of this investigation.

5.4.1 Experimental data details

The experiments were conducted on four types of data volumes: daily, weekly, monthly and randomly selected data from the months of July 2015 to October

2015. The analysis has been performed on the historical data of case study-1 to reveal hidden patterns through the X-means clustering. Day analysis was performed on 17^{th} July 2015 TUs, week analysis performed from 17^{th} July to 23^{rd} July for 5 working days (excluding weekend) and month analysis done for 22 days since weekends have not been included in the analysis. The random analysis was conducted on the randomly selected TUs from July 2015 to October 2015. TU data details are given in Table 5.1.

TABLE 5.1: Experimental data details

No of Days	Total number of TUs	Operating TUs	Description
1 Day	731	723	17^{th} July, 2015
1 Week, (5 days)	3655	3615	17^{th} to 23^{rd} July 2015 (except 18^{th} and 19^{th})
1 Month, (21 Days)	15351	15178	17^{th} July to 14^{th} August 2015 (except weekends)
3 Months, (71 Days)	39088	38591	17^{th} July to 23^{rd} October 2015 (except weekends)

5.4.2 Clusters analysis via radar representation

The analysis began with one day TU data and then moved to experiment on more days (tabulated in Table 5.1). The optimum number of clusters have been chosen based on the BIC criterion [79] determined to perform the clustering. Initially the data must be partitioned into at least two groups, where one group represents faulty and the other signifies non-faulty. Then BIC scores have been calculated for the whole dataset, the two groups separately and then the scores between them compared. This process continues until the current partition achieves better scores than the previous splitting. It has been found that the best BIC score was obtained with 'six' number of clusters. Thus, six is considered as the ideal number of partitions to represent the different TU patterns and to perform the clustering experiments. The validation of cluster compactness has then been determined

statistically, where the cluster numbers have been varied from two to ten for all the methods used for comparison and is explained at the end of this section.

The X-means clustering algorithm was tested on approximately forty thousand TUs for case study-1 over a period of four months (July to October) in 2015 as tabulated in Table 5.1. These experiments began looking at daily data and then further conducted on weekly, monthly, and randomly selected data and results are described in subsections below.

5.4.2.1 Terminal unit analysis for a day

On a day altogether seven hundred twenty three (723) TU's behavioural patterns are captured into six clusters. These TUs distinct behaviour visualization has been displayed via six different radar graphs in Figure 5.3. It has been observed from the radar graph that mostly TUs, which are grouped in same cluster depicts similar behavioural pattern. In each radar the nearest TU data point from each cluster centroid is plotted in the below radar graphs (Figure 5.3) to make an assumption about the individual group or cluster operation.

Figure 5.3 shows six different cluster radar that represent six different types of TU behaviour in terms of temperature and power variations inside the building of case study-1 over a 24hr period. The heating trends are represented by pink colour and cooling by blue colour. It is found from the radar that the heating trend is captured in cluster-3 and cooling trends are captured in the rest of the clusters (cluster-0, 1, 2, 4, and 5). TUs belonging to cluster-0 achieve their set point and the feature values lie in third axis (FC_3) of the radar. It illustrates that the calculated feature area for that TU is within the goal achieved (GA) to event end (EE) events for the temperature curve. Therefore these TUs which are in cluster-01 exert little power effort and stay within their dead band as described in Section 4.2. Cluster-1 represent TUs whose features falls in the third axis (FC_3),

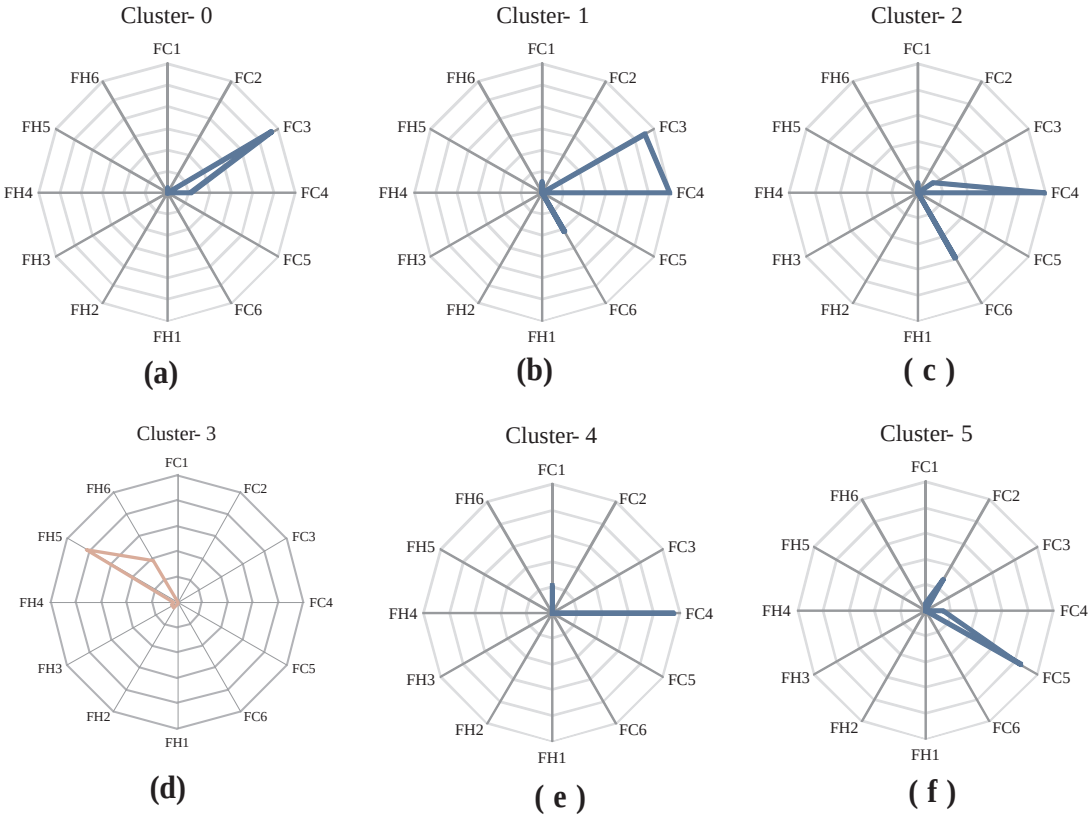


FIGURE 5.3: (a) - (f) Six radars to represent different types of TU behaviour for a day.

indicating GA to EE state as well but, those TUs required average power (e.g., values are in FC_6 axis) to reach their set point. Thus, TUs belong to cluster-1 use more power than the TUs are in the cluster-0. The TUs of cluster-2 behave similar to cluster-1 but these TUs demand higher power levels to reach the set point within the buildings operational hours. TUs of cluster-3 capture the area under response delay (RD) to GA (FC_5) is large for both the temperature and power curves, which depicts more heating power is required, and still the set point is not achieved by the TU during the working hours. The feature values of cluster-4 fit in first axis (FC_1) implies longer time requirement to increase the control temperature of those units. Thus, area under the event start (ES) to RD state for both the temperature and power curves that the TUs take longer and are unable to achieve the required set points. In case of cluster-5, both the temperature and

power curve areas lie under the RD to GA events. This depicts that though the power demands is high still the control temperature unable to reach the set point or unable to achieve the goal. This exhibits 'badly behaving' TU patterns. By this approach, all the TU behaviours are represented and categorised as faulty or non-faulty patterns.

TABLE 5.2: Discovered cluster pattern description of daily TUs

Cluster Number	Description of Clustering Pattern
C-0	TUs that show both heating and cooling characteristics. The desired control temperature is achieved with low power effort, and is done in normal building operating hours.
C-1	These are cooling TUs where the control temperature is achieved within the desired band with medium or average power effort.
C-2	Cooling TUs where control temperature is achieved but with a high-power effort.
C-3	TUs showing control temperature hunting patterns along with medium to high power effort and continuation of this pattern outside operational hours.
C-4	TUs where the control temperature does not achieve the desired set point (out by up to 5 degree C) with high power effort.
C-5	TUs where the control temperature does not achieve the desired set point (out by up to 10 degree C) with high power effort.

After performing the X-Means clustering for daily TU data, it is partitioned into six different groups based on their properties. These clusters revealed six distinct patterns to represent the behaviour of daily TUs of the case study-1. The distinct TU behaviours that are obtained from clustering results are inferred in Table 5.2. The aim of this clustering algorithm is to determine specific TU behaviours. It can provide an insight into the TU characteristics without any prior knowledge to identify faulty and non-faulty trends. Each cluster is labelled as C-0, C-1, C-2, C-3, C-4, and C-5. This labels are then use to classify the TUs for automatic fault detection and diagnosis purposes.

5.4.2.2 Terminal unit analysis for week, month, and randomly selected data

Following these, further experiments were conducted on the TUs for week, month, and randomly selected data over a specific period (mentioned in Table 5.1) to investigate the feasibility, and robustness of the proposed clustering method. Figure 5.4 shows the cluster analysis of TU behaviours during week days, Figure 5.5 for a month, and Figure 5.6 for randomly selected TUs. Repeated TU behaviour is found during the clustering experiment on weekly, monthly, and randomly selected data. TUs of cluster-0 of Figure 5.3, cluster-0 of Figure 5.4, and cluster-3 of Figure 5.5 are found to have similar heating and cooling trends, where the goal or desired temperature is achieved with very low power demand. Likewise, the TUs of cluster-1 of Figure 5.3, cluster-2 of Figure 5.4, cluster-0 of Figure 5.5, and cluster-0 of Figure 5.6 follow similar trends where the set-point is achieved with average power effort. In the case of TUs from cluster-2 of Figure 5.3, cluster-5 of Figure 5.4, and cluster-2 of Figure 5.6 higher power effort are needed to reach the goal or specific set-point. The hunting behaviour of TUs is found with high power consumption during operational and out of operational hours in cluster-3 of Figure 5.3, cluster-1 of Figure 5.4, cluster-1 of Figure 5.5, and cluster-3 of Figure 5.6. Likewise the daily TU data of cluster-4 of Figure 5.3 is similar to cluster-3 and cluster-5 of Figure 5.4 and Figure 5.5, where set-point is not achieved but power is still consumed during out of operational hours. Similar TU behaviours e.g. cluster-5 of Figure 5.3 is found respectively in cluster-4, cluster-4, and cluster-5 of Figures 5.4, 5.5, and 5.6. It is noted that the patterns found in cluster-2 of Figure 5.5 and cluster-4 of Figure 5.6 are very similar to each other, where the set-point is achieved and both captured heating trends. Another two similar heating patterns of TUs are found in cluster-1 and cluster-3 of Figure 5.6, both achieved set-point using average to high power effort.

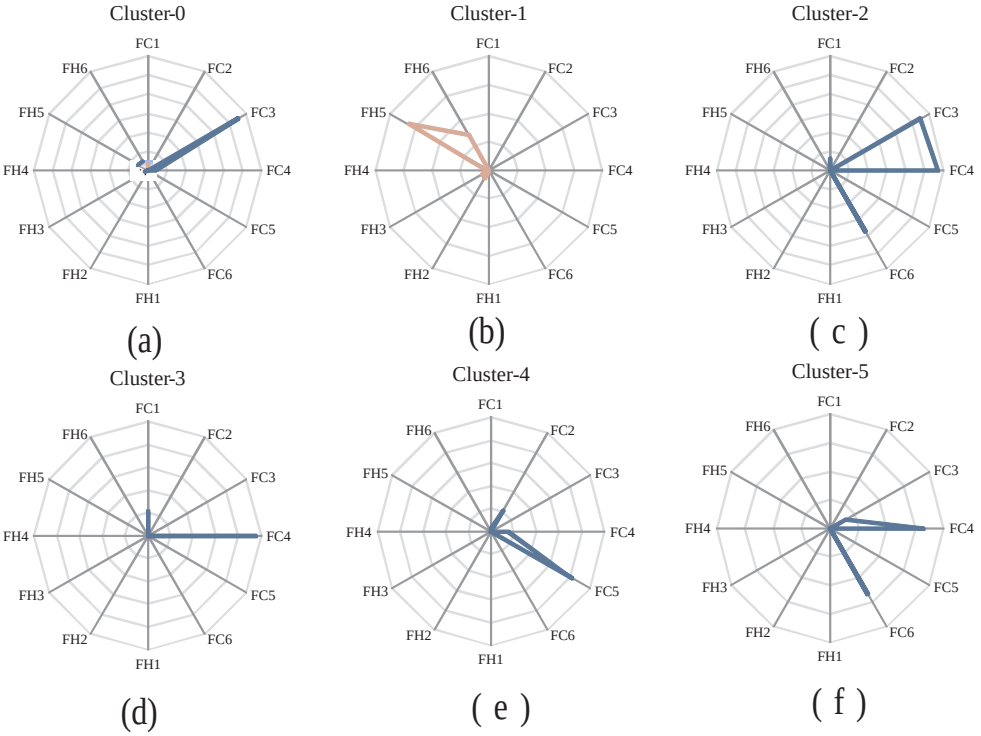


FIGURE 5.4: (a) - (f) Six radars to represent different types of TU behaviour for a week.

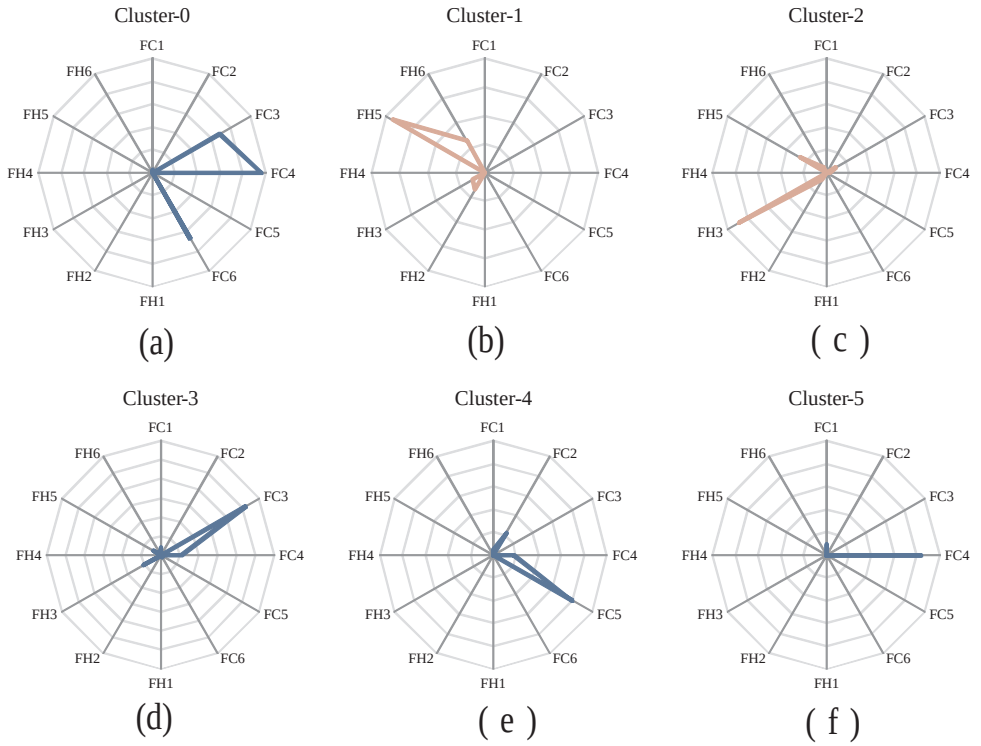


FIGURE 5.5: (a) - (f) Six radars to represent different types of TU behaviour for a month.

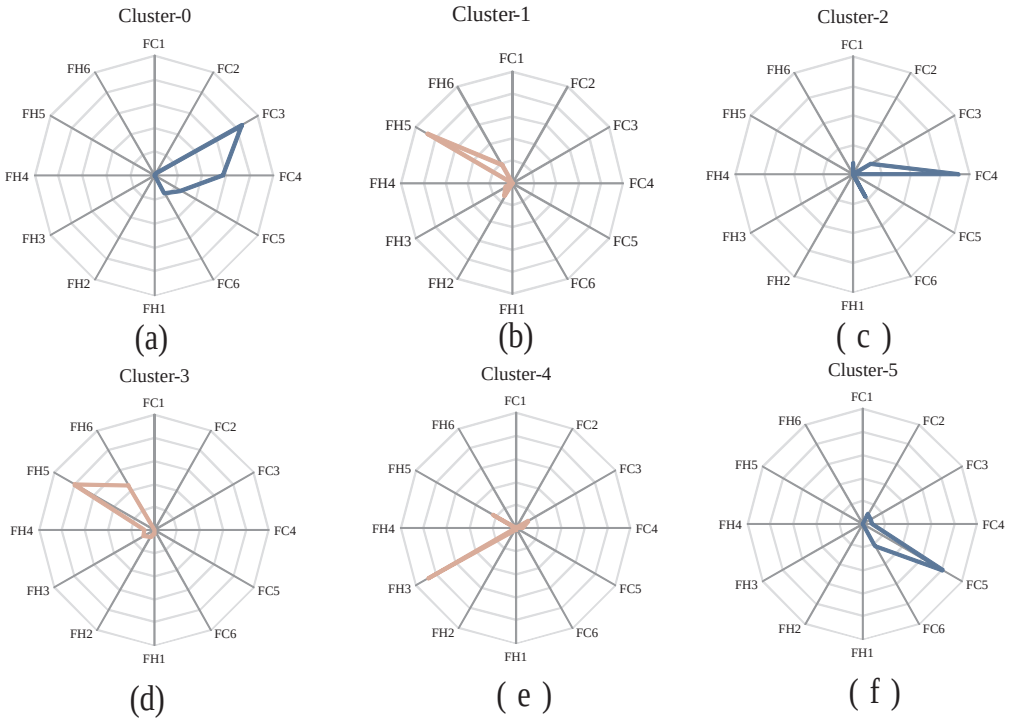


FIGURE 5.6: (a) - (f) Six radars to represent different types of TU behaviour of randomly selected data from July to October 2015.

It can be concluded after the analysis that six radars reflect similar type of heating or cooling patterns for each of the daily, weekly, monthly, and randomly selected TU clustering experiment. The heating or cooling trends may differ depending on the weather conditions where control temperature varies, but it captures the specific swing pattern of temperature and power in a cluster every time. Once the radar visualization experiment has been done for various range of TU data, it has been further mapped with the actual individual TU operation for that day. This relationship between the radar representation and the actual TU behaviour are discussed in detailed now.

5.4.3 Terminal unit's performance mapping with radar graph

Each radar graph represents an individual type of TU behaviour and each of its axes represents a separate feature as described earlier. The six cooling features are represented by $(F_{C_1} - F_{C_6})$ on the right hand side axes of the radar and the

six heating features are represented by ($F_{H_1} - F_{H_6}$) on the left hand side axes respectively. Figure 5.7 shows an example of "good" behaving TU in heating mode as the pink lines are on the third left axis (F_{H_3}) this signifies the control temperature lies within the desired heating and cooling setpoints. Also, the pink lines are between the fourth (F_{H_5}) and fifth (F_{H_6}) axes of the radar signifying the respective medium power demand to maintain that temperature within set the goal area. This TU example only demand heating power thus the graph shows the heating power in pink colour.

Figure 5.8 shows another example of a "well-behaved" TU in cooling mode as the blue lines are on the third right radar axis (F_{C_3}) showing that the control temperature achieves the desired set point by demanding very little amount of cooling power.

Examples of three well-known but different TU faults: saturation, hunting and high temperature error patterns are now described. Figures 5.9 and 5.10 show saturation and high temperature error behaviours indicating that for high proportion of time during the day, the valve or damper is open at a maximum level. Figure 5.9 shows hunting behaviour and shows how much the control temperature fluctuates over a day. Further, it can be seen from Figure 5.10 and Figure 5.11 that the TUs are continue to operate even of out of normal hours displaying further energy wastage and unnecessary usage.

This behaviour also indicates a high degree of on-ness, that is the proportion of time that a TU had any heating or cooling demand over a 24-hour period. From the radar representation in Figure 5.9, it can be be seen that the curve lies on the second left axes (F_{H_2}) of the radar graph indicating that the control temperature spends more time in the RD event. In the case of Figure 5.10, the curve lies on the first right axes (F_{C_1}) of the radar graph indicating that the control temperature spends more time in the ES event, whereas for Figure 5.11 the curve mostly lies

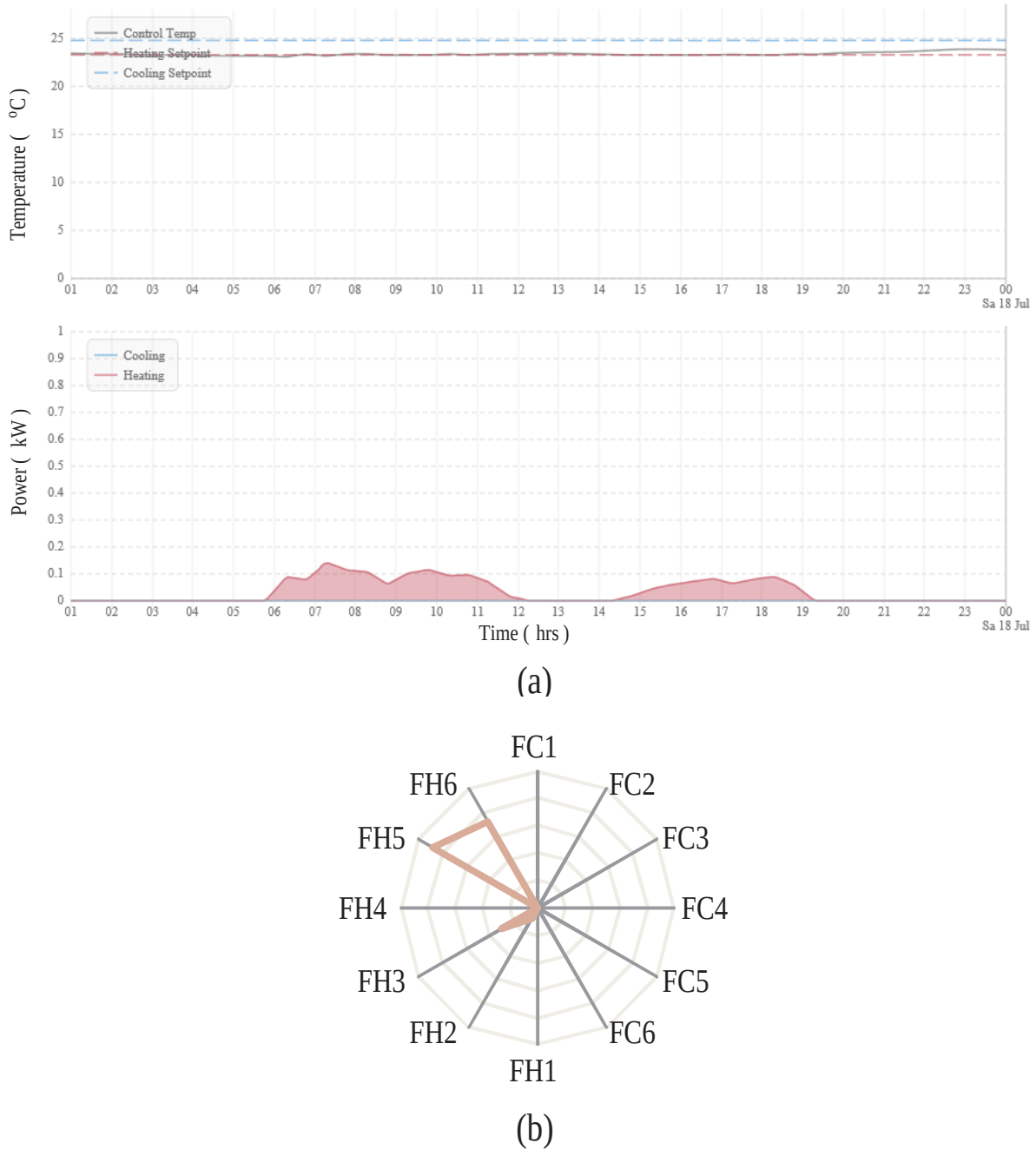


FIGURE 5.7: A heating example of a "well-behaved" TU. (a) actual temperature and corresponding power consumption, (b) TU feature representation via radar graph.

in RD to GA event for both the temperature and power curves. These radar representations explain the actual TU behaviour as shown in the top section of each of the figures. Moreover, all of these TU with high average power also have high temperature errors that is control temperature deviates highly from the set point,

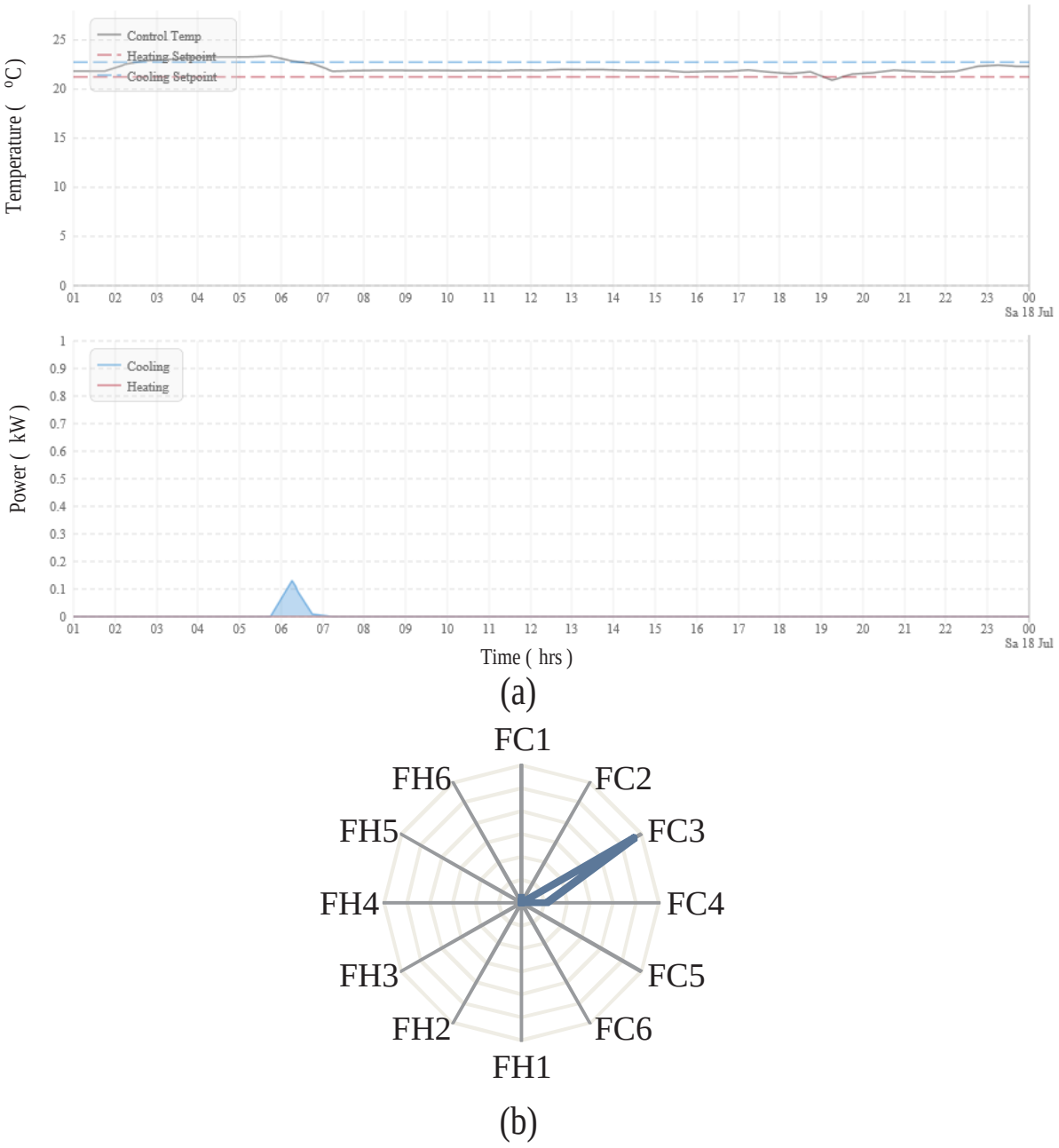


FIGURE 5.8: A cooling example of a "well-behaved" TU. (a) actual temperature and corresponding power consumption, (b) TU feature representation via radar graph.

indicating that they need to look after immediately for further investigation and are not only poorly controlling the temperature, but also consuming a relatively large amount of energy too.

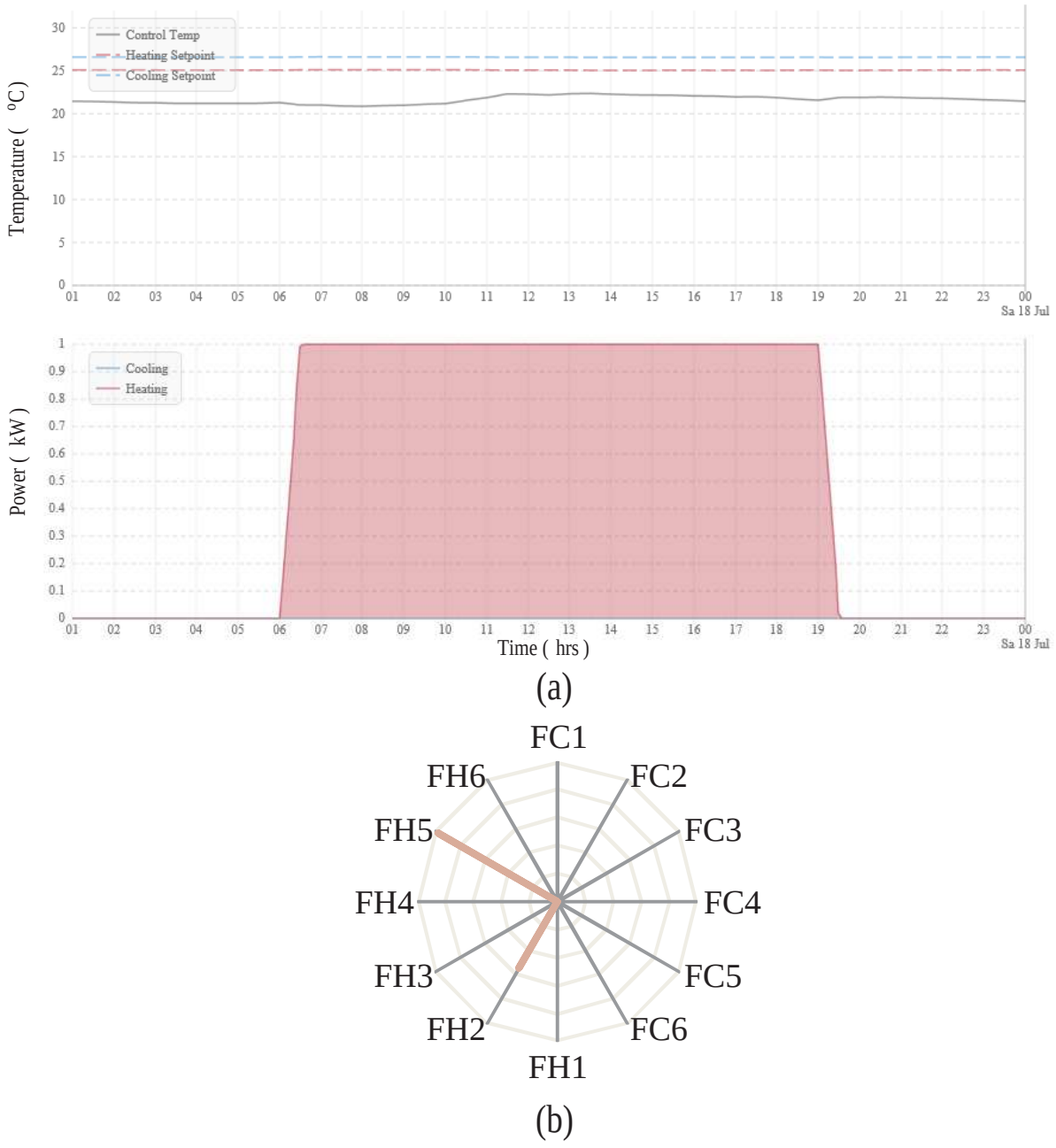
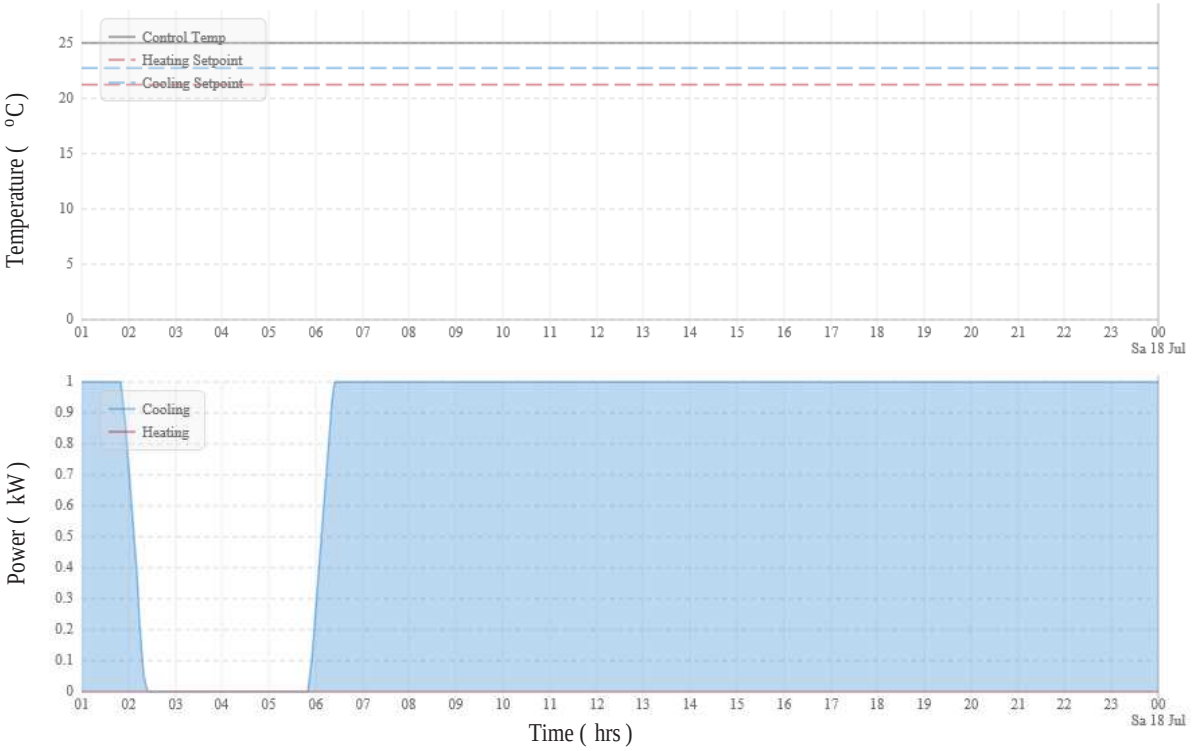


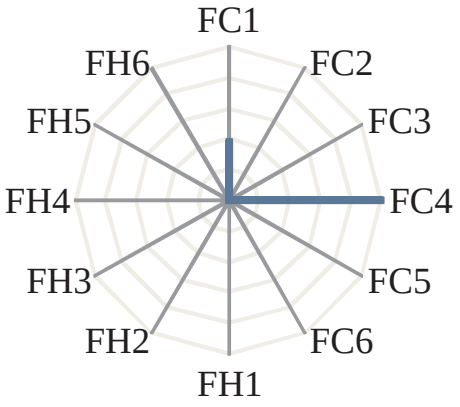
FIGURE 5.9: An example of a "badly-behaved" TU (a) actual temperature and corresponding power consumption showing heating saturation with high temperature error, (b) TU feature representation via radar graph.

5.4.4 Outlier analysis

The effect of outliers on clustering is examined here and those TUs, which are far from their own cluster centroid are investigated. The outlying TU datapoint might have dissimilar behaviour from the other TUs within that cluster and can affect the



(a)



(b)

FIGURE 5.10: An example of a "badly-behaved" TU (a) actual temperature and corresponding power consumption showing cooling saturation with on-ness, (b) TU feature representation via radar graph.

entire clustering outcomes. From the proposed clustering outcomes, three "well-behaved" and three "badly-behaved" clusters have been found. Therefore, the aim of this outlier analysis is to discover the suitability of TUs within its cluster. For example, a TU is in a "well-behaved" cluster, where control temperature and

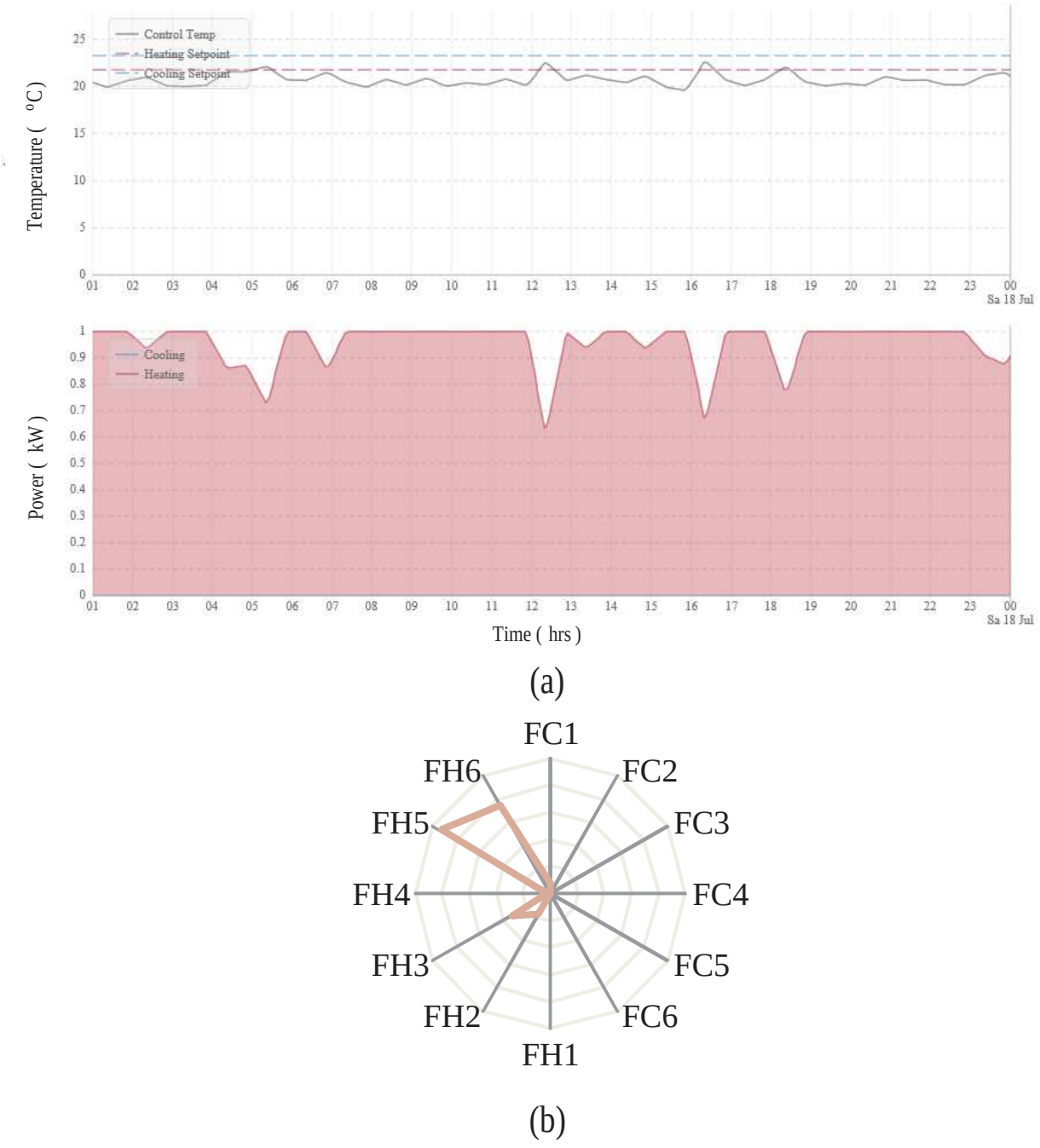


FIGURE 5.11: An example of a "badly-behaved" TU (a) actual temperature and corresponding power consumption showing heating hunting with on-ness , (b) TU feature representation via radar graph.

power demand are always desirable. Though the TU has good temperature control in it has a higher power demand than others from that cluster. That might affect the whole cluster nature and would consider as anomalous behaviour for that TU group. Thus, the distance between each data point from their cluster centroid is

then measured for all six clusters. As different number of datapoints consist of different feature values that varies the compactness within clusters. Therefore, this outlier analysis aims to refine the TU behaviour analysis by identifying those TUs which are far from their centroid. A threshold is determined to select meaningful TUs in every cluster and if found beyond threshold can be considered as outlier to look further. Experimentally, the threshold value is decided from cluster mean and standard deviation for each cluster separately. Once the distance value of a data point from the cluster centroid is greater than a measured threshold that is: $(2 \times \textit{standard deviation} + \textit{mean})$, then it is considered an outlier behaving TU.

This outlier finding experiment has been executed for daily TUs and the results are presented in Figure 5.12. The plot displays outliers in each cluster based on their own measured threshold. The y-axis represents the number of clusters (from cluster-0 to cluster-5) and the x-axis represents the data points distance from their centroid. All the dotted points represent the total number of TUs, where the red dots (to the right) depict the TUs beyond the considered threshold based on the clusters standard deviation and mean, which represents outliers or anomaly behaving TUs in that cluster. Although, it is not completely possible to determine if an outlying point represents a truly faulty or non-faulty TU. But, those are not exactly similarly behaving TUs to others in that cluster. Furthermore, this outlier analysis is verified by resident expert building engineers at Demand Logic to ensure the results are meaningful.

The clustering outcomes further statistically validated and compared against other algorithm using the Silhouette indexing [84] and Davis Boulding indexing [83] methods as described in Section 5.3.4.

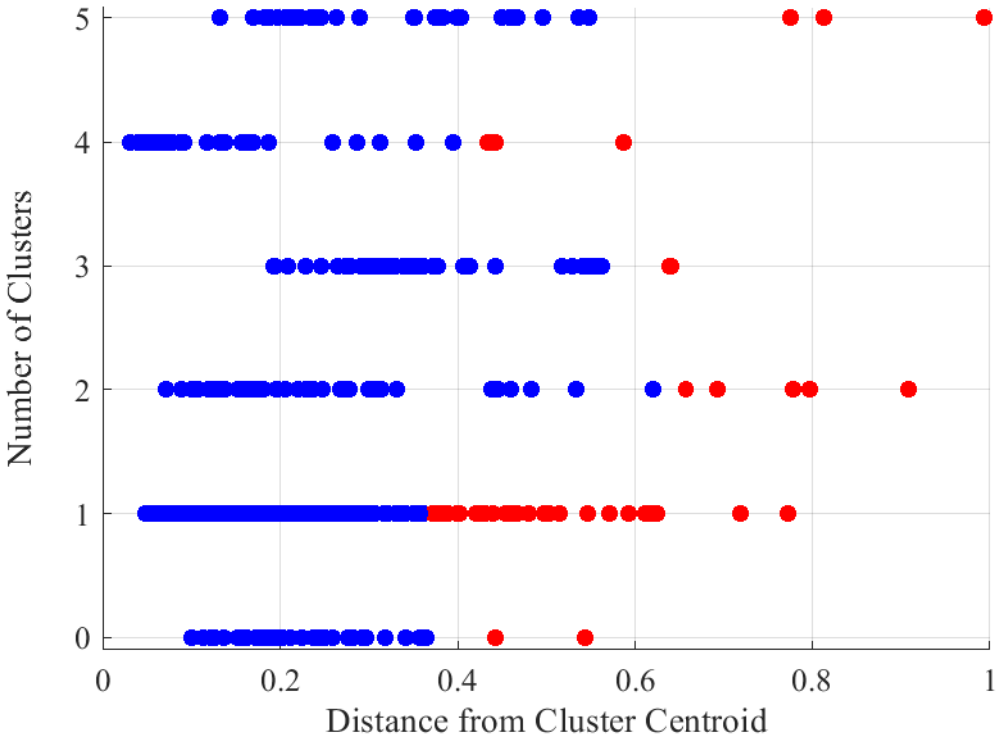


FIGURE 5.12: Cluster wise outlier analysis for daily TU data.

5.4.5 Cluster validation and comparison

This TU pattern discovery process is completely unsupervised as label information is not available priori. Thus, the internal cluster evaluation criteria are used to validate the clustering results and assess compactness of those groups as described in Section 5.3.4. Two statistical criteria: Davies-Bouldin (DB) [83] and Silhouette Indexing (SI) [84] have been employed here. The proposed X-means results are compared with other two well-known clustering algorithms: hierarchical clustering and k-medoids [85] and obtained better performance over other methods. The validation results is summarized in Figure 5.13. The comparison and validation result across daily, weekly, monthly and randomly selected TU data are included here.

Figure 5.13(a) demonstrates the DB indexing results of X-means by varying the

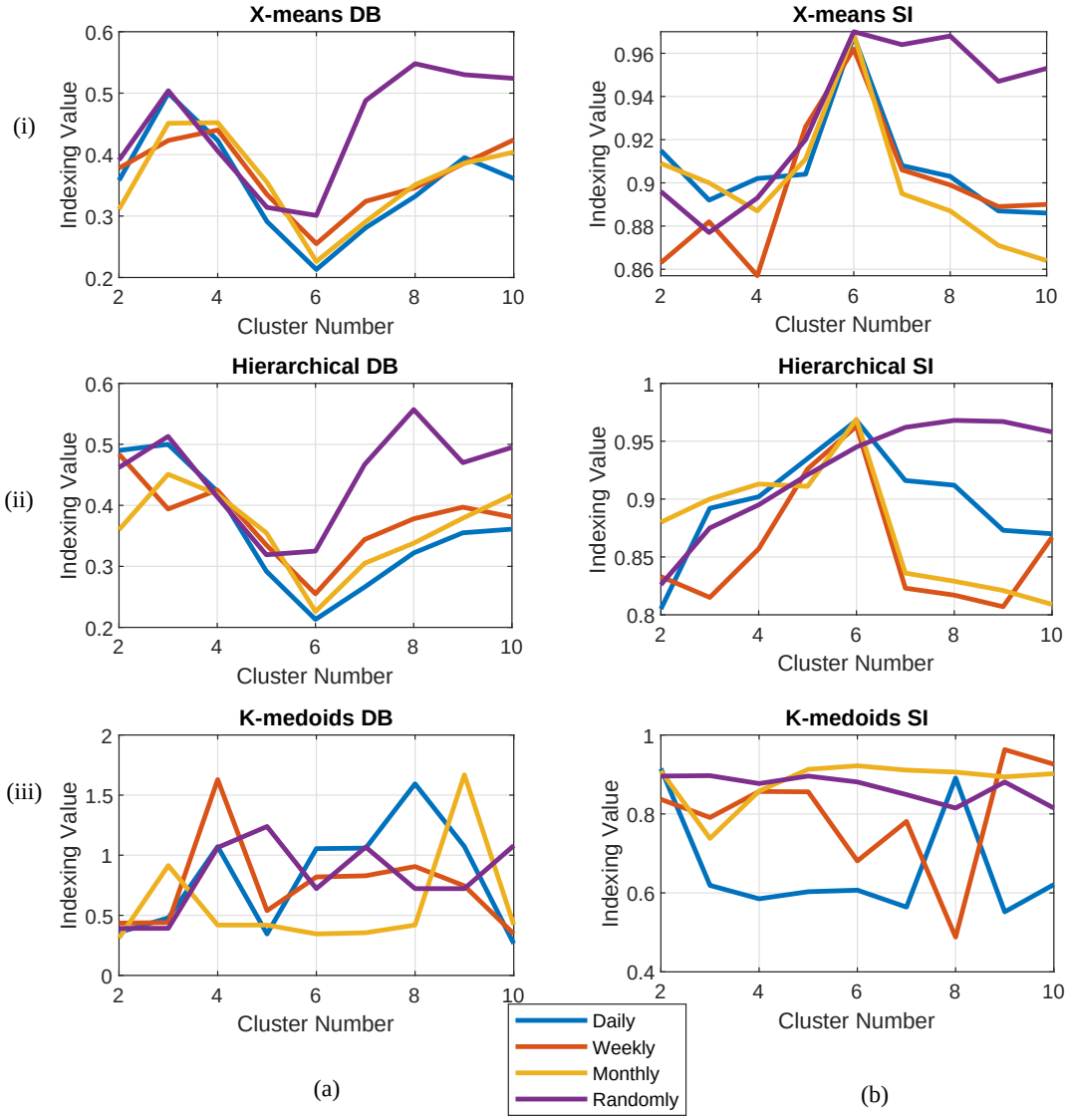


FIGURE 5.13: Clustering validation and comparison.

cluster number 2 to 10 and is compared with hierarchical and k-medoids algorithms. As the lowest DB value is preferred, it is seen from the Figure 5.13(a) that lowest DB value is achieved when the cluster number is six for all the cases of daily, weekly, monthly, and randomly selected TUs for X-means. Similarly, lower DB values are found in the hierarchical clustering for all the cases, but values are higher than X-means. However, the heavy fluctuation of DB values is shown in the case of k-medoids algorithm. Therefore, the lower value of DB in X-means for

cluster number 6 shows a better compactness of the clusters than the hierarchical and k-medoids clustering methods proving the model validation for this TU clustering process.

Subsequently, Figure 5.13(b) shows the SI indexing results for X-means, hierarchical and k-medoids algorithms. The clusters with high SI index are well matched to its own cluster. It is clearly seen in the Figure 5.13(b) that the SI index is high for X-means on daily, weekly, monthly, and randomly selected TU data, when the cluster number is six, whereas hierarchical clustering achieved high SI indexing for cluster number six as well for all cases except the randomly case. Again, in the k-medoids algorithm, SI values are fluctuating for all the cases, therefore it is hard to discover specific cluster numbers for the data set and to define their compactness within clusters. These internal clustering validation results shows that the selected optimal number of clusters is appropriate and achieved suitable indexing outcomes consistently for X-means than other clustering algorithms. These clustering results were further verified with the help of expert building engineers at Demand Logic in moving the research direction towards the thesis goal of automatic fault detection and diagnosis tool for buildings.

5.5 Conclusion

In this chapter, a novel feature extraction technique for clustering has been employed to discover distinct TU behaviours. With the application of an unsupervised learning technique, a number of different cluster patterns have been analysed and identified that help in the identification of different TU behaviours. These results were thoroughly analysed to form concrete knowledge and provide an insight in the direction of different TU behaviour identification. A strong visualisation with respect to feature extraction and the relation between faulty and non-faulty TU has been drawn through the radar graph. This could give a clear conception

of all the possible different behaviours inside the building. To make a robust algorithm this method was repeatedly tested over more periods such as week, month, and randomly throughout four months including the summer and winter for historic building data. This analysis also validated statistically and manually with help from expert building engineers.

The motive of the proposed clustering is to identify significant TU patterns where these distinct clusters are used to classify normal and abnormal TUs. Further the identified of abnormal TUs will be taken care by the building engineer for resource optimisation. Subsequently, based on the obtained clustering behaviours, categorical label assignment has been enacted to create an automated TU classification system explained in the next chapter.

Chapter 6

Supervised learning

"He who struggles is better than
he who never attempts."

-Swami Vivekananda

6.1 Introduction

Remote and automatic identification of faulty terminal units (TUs) have significant impact on building services, maintenance and energy consumption [86]. It needs appropriate information about the TU data from the building expert system to support supervised learning method to act efficiently for anomalous activity recognition. A building's HVAC terminal unit producing huge amounts of data and the behaviour of these units are diverse in nature. Therefore, recent machine learning developments have the potential to learn and predict once the appropriate data knowledge is available [5]. This has been examined here through a realistic approach for real building data.

In this chapter, supervised learning (SL) or classification method is considered as a sequel process for creating the full AFDD system in this thesis. This SL is applied to classify faulty and non-faulty TUs by the discovered knowledge from the previous chapter. These behaviour findings have been accomplished in Chapter 5, employing clustering techniques and six types of behaviour were discovered from the case study building in London. The behaviour information was then verified by the expert building engineers and are now used here for TU labelling and recognition purposes. The classification has been performed by training and testing the algorithm repeatedly to understand the trends. This method can distinguish faulty TU pattern automatically and alert the building engineering to investigate the equipment depending on the classification outcomes. In this AFDD research supervised learning plays a vital role towards early diagnosis of faults, appropriate follow-up and maintenance that leads to improved building performance and controlling significant energy loss.

6.2 Proposed classification methods

The proposed framework for classification is shown in Figure 6.1. It shows the work flow of the whole process involved in this chapter, and comprises two stages: unsupervised and supervised learning. Firstly, the vast data is held in a secure cloud and further applied feature extraction method, subsequently clustering is employed to group the similar patterns and dissimilar patterns into other groups based on their characteristics, as explained in the previous chapter. Here, the clustering outcomes have been used for data labelling to classify the TU data. The Multi-class support vector machine (MC-SVM) is used as classification model for AFDD purpose. In order to find out optimum model selection, another two

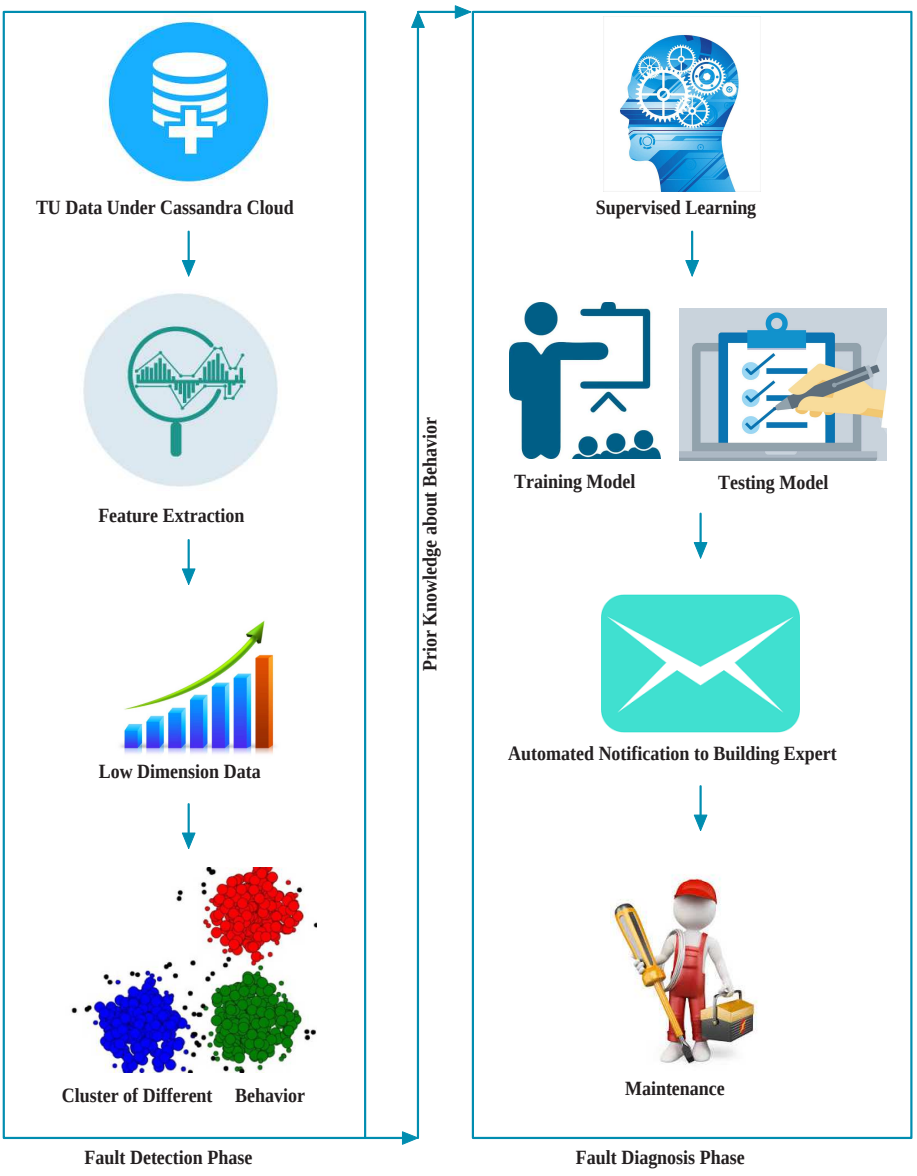


FIGURE 6.1: Computational steps involved in proposed methodology.

classification algorithms have investigated: K-nearest neighbour (KNN) and multi-layer perceptron (MLP) for results comparison and validation through precision and recall are explained below [87].

The experiment has been carried out using Matlab R2016b tool on an Intel(R) Core(TM) i5 processor@3.30 GHz 130 running Windows 7 Enterprize 64-bit operating system with a 7856-MB NVIDIA Graphics Processing Unit (GPU).

6.2.1 Multi-class support vector machine

Support vector machine (SVM) is a well-known binary classifier modelled here as Multi-class support vector machine (MC-SVM) to deal with a greater number of classes. Thus, the classes are discriminated by a hyperplane as shown in Figure 6.2. This figure represent a hyperplane as a line in two-dimensional space, which is to divide the data into two parts. The model learns the hyperplane to create a good margin, where this separation is larger for both the classes, is assumed the perfect decision boundary to data partition. This is performed by transforming the problem with linear algebra known as the kernel function. There are many kernels developed but specifically used depending on the data distribution. In this study, two types of kernel functions:liner kernel and quadratic kernel, have been chosen for examine both the linear and non-linear data types.

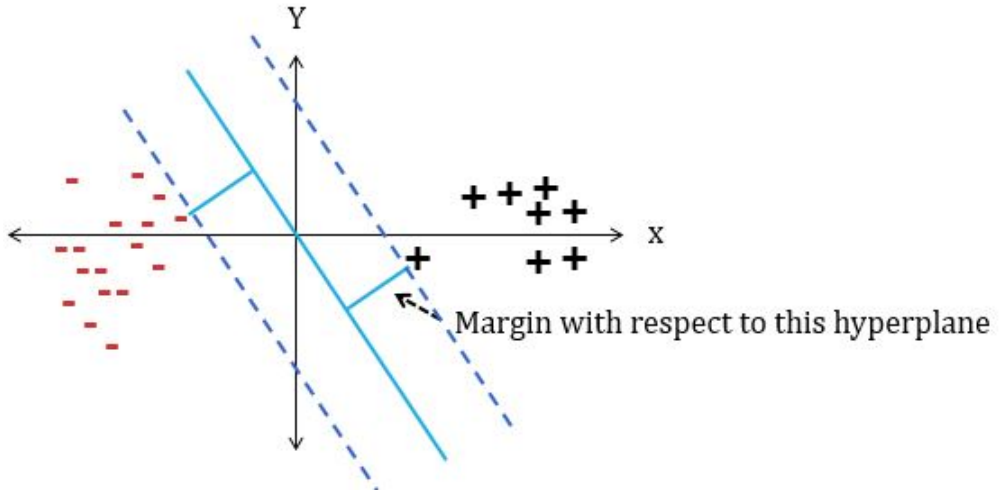


FIGURE 6.2: Decision boundary of support vector machine.

MC-SVM [88, 89, 90] is employed on the extracted TU feature by optimizing the distance between support vectors (TUs from different groups) defined as,

$$\begin{aligned} \min_{w_m \in H, \xi \in R^l} \quad & \frac{1}{2}\sum_{m=1}^{k} w_m^T w_m + C\sum_{i=1}^{l}\xi_i \\ \textit{subject to} \quad & w_{y_i}^T\varphi(x_i) - w_t^T\varphi(x_i) \geq 1 - \delta_{y_i,t} - \xi_i \\ & i = 1, ..., l, t \in 1, ..., k \end{aligned} \tag{6.1}$$

In (6.1) a set of training patterns is denoted by $(x_1, y_1), ..., (x_l, y_l)$ of cardinality l, where $x_i \in R^d$ and $y_i \in 1, ..., k$, $w \in R^d$ is the weight vector, $C \in R_+$ is the regularization constant, and φ is mapping function which projects training pattern into a suitable feature space H that allows for nonlinear decision surfaces. The constraints $\xi_i \geq 0, i = 1, ..., l$, are implicitly indicated in the margin constraints of (6.1) when t equals y_i. The final decision function is defined as,

$$argmax_m f_m(x) = argmax_m w_m^T\varphi(x) \tag{6.2}$$

Whereas in (6.2), $\delta_{i,j}$, is the delta (defined as 1 for $i = j$ and as 0 otherwise). In addition, Equation (6.1) focuses on classification rule (6.2) without any bias terms. A non-zero bias term can be simply exhibited by adding an additional feature to each x. Therefore, different categories of data are classified by solving this decision function and the results are analysed in the following section. A pseudo code for the proposed work is presented in the Algorithm 1.

6.2.2 K-nearest neighbor

In the classification study, K-nearest neighbour (KNN) plays a vital role. It is a simple and effective distance based non-parametric pattern recognition algorithm [91, 92]. KNN select the output class based on the majority voting received for the pattern. It begins with a random initialized k value (it can be varied as

Algorithm 1 Pseudo code for proposed AFDD

Require: Temperature and power feature of TUs = $\{T_1, T_2, ..., T_n\}$

1: Fault detection stage:
2: $Maximum\ cluster\ number = Max$
3: $K = BIC\ score\ for\ previous\ model$
4: $K_{max} = BIC\ score\ for\ current\ model$
5: **for all** $Max = 2$ *to* 10 **do**
6: *Assign initial values for means* $= \mu_1, \mu_2, ..., \mu_{Max}$
7: *Assign each TU to the cluster which has closest mean* $\Leftarrow$ *using the Eq.* 5.1
8: *Update means*
9: *Calculate BIC score for current model* K_{max} $\Leftarrow$ *using the Eq.* 5.2
10: **if** $K > K_{max}$ **then**
11: **return** *Go to line* 5
12: **else**
13: *Max is considered as final cluster number & exit loop*
14: **end if**
15: **end for**
16: *Six groups of labelled patterns* $= \{C0,\ C1,\ ...,\ C5\}$ *are produced by* $X-$ *means*
17: Fault diagnosis stage:
18: *Train* $MC-SVM$ *model by* 10%, 20%, *and* 30% *data using objective function*
19: *Make the prediction on rest of TUs*

per the experiments). If K = 1, then the case is simply assigned to the class of its nearest neighbor. Depending on the k values it searches that number of nearest points and assigns them into the class which the nearest points belong to. K-nearest neighbors measured by a distance metric or proximity measure. Distance metric is a function that defines a distance between each pair of elements of a dataset. The goal is to learn from a similarity function that measures how similar or related two objects are. Here, four types of distance function are investigated for the similarity checking purpose.

Euclidean distance is a straight-line distance measure between two points in Euclidean space [93]. This is the best proximity measure when data is dense or continuous and presented as,

$$d_E(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{6.3}$$

Where, d_E is the sum squared distance between two data points of q_i and p_i from the whole n number of data points.

Jaccard distance is a measure of how dissimilar two sets are. It is obtained by subtracting the Jaccard coefficient from 1 [94]. The Jaccard coefficient measures similarity between finite sample sets, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union. Where, d_J is the coefficient distance between two set of P and Q.

$$d_J(P,Q) = 1 - J(P,Q) = \frac{|P \cup Q| - |P \cap Q|}{|P \cup Q|} \tag{6.4}$$

City-block distance or **Manhattan** examines the absolute difference between coordinates of a pair of objects [95]. Where, d_C is the absolute difference between the coordinates of p_i and q_i from the whole n number of data points.

$$d_C(p,q) = ||p - q||_1 = \sum_{i=1}^{n} |p_i - q_i| \tag{6.5}$$

Minkowski distance measures different orders between two objects with three variables [93]. Minkowski distance (d_M) is typically used with p being 1 or 2, which correspond to the Manhattan distance and the Euclidean distance, respectively. In the limiting case of p reaching infinity.

$$d_M(P,Q) = (\sum_{i=1}^{n} |p_i - q_i|^x)^{\frac{1}{x}} \tag{6.6}$$

6.2.3 Multilayer perceptron

A multilayer perceptron (MLP) [2] is a class of feedforward artificial neural network employing a nonlinear activation function. It is able to distinguish data that are not linearly separable. It generally consists of three layers: input layer, hidden layer, and the output layer shown in Equation (6.3). MLP generally uses back-propagation algorithm for calculating the gradient, which needed for weight adjustment throughout backwards layers in the network to minimize the error presented as,

$$y = \varphi(\sum_{i=1} w_i x_i + b) \tag{6.7}$$

It reoccurs until the error will go negligible and finally the model training will be completed. In the hidden and output layer it uses a non-linear activation function. It mainly uses the sigmoid function (S(x)) shown as,

$$S(x) = \frac{1}{1 + \exp^{-x}} = \frac{\exp^{x}}{\exp^{x} + 1} \tag{6.8}$$

Where w denotes the vector of weights, x is the vector of inputs, b is the bias and φ is the non-linear activation function.

Backpropagation (BP)- The error (ξ) will back propagate through the each layer using,

$$\xi(n) = \frac{1}{2} \sum_{j} e^2(n) \tag{6.9}$$

$$\exp_j(n) = t_j(n) - y_i(n) \tag{6.10}$$

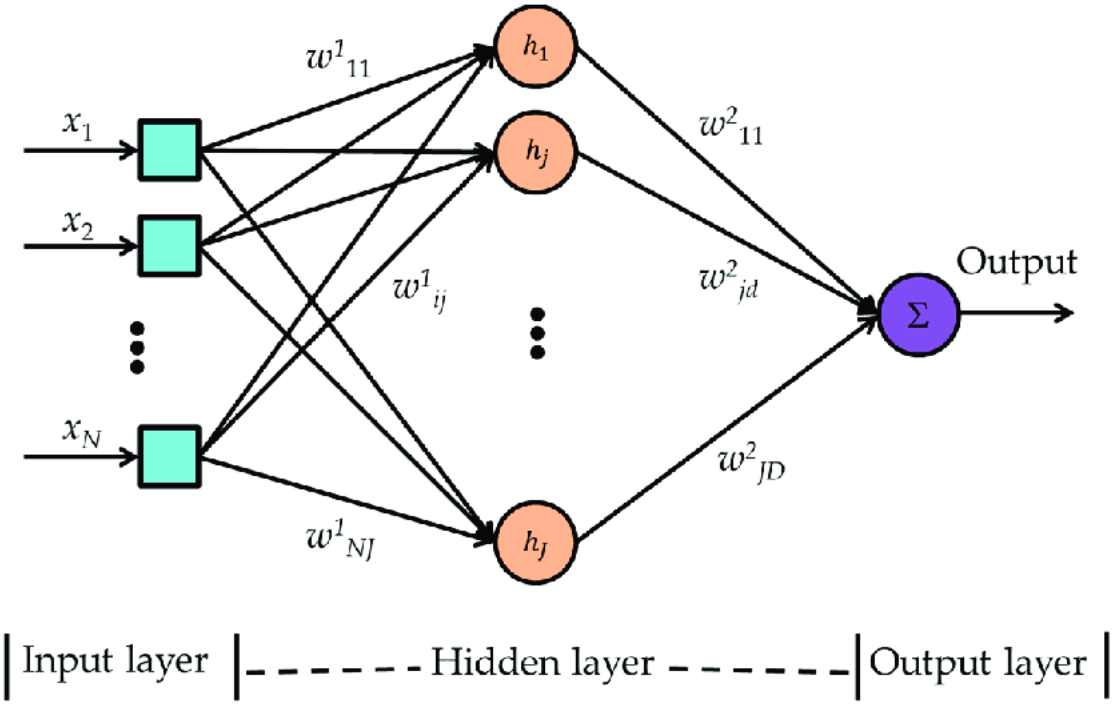


FIGURE 6.3: Computational steps involved in multi-layer perceptron. [2]

Where, the error is calculated through the e, which is actually the difference between the target value (t_i) and the produced value (y_i) shown in (6.9) and (6.10).

6.2.4 Classification performance evaluation

After performing classification the obtained results are evaluated through well-known statistical performance metrics, precision (Pr) and recall (Re) for calculating the accuracy of the system. Precision calculate the percentage of truly predictive TUs among all the positively detected TUs in the dataset and recall measures the relevant outcomes which are detected among the total dataset shown as,

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{6.11}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{6.12}$$

The performance measurement is based on the number of true positive (TP), false positive (FP), and false negative (FN). TP is the total number of correctly labelled TUs, whereas FP is the total of incorrectly assigned TUs, and FN is the total of not labelled as should be belonging to particular classes.

6.3 Classification result analysis

In the result analysis different classification methods have been applied on the TU data of the building-1 (described is Section 3.3) for daily, weekly, monthly, and randomly basis. The experimental data have been detailed in the previous chapter (Scetion 5.4). In this phase, classification has been performed by employing MC-SVM which is considered a baseline classifier in this pattern classification problem. Here the dataset is categorized and labelled into six different classes for diagnosing diverse behaviour of faulty and non-faulty TUs (tabulated in Table 5.2). The MC-SVM is employed for performing the classification task and for result comparison KNN and MLP have been investigated and results are included here.

Here, the experiment is conducted by dividing the TU dataset in 10%, 20%, 30% as training data, and 90%, 80%, 70% respectively as testing data. Where the observation for each training was also tabulated as number of objects (NOB). Also, the experiment has began with daily TU data and expand to weekly, monthly, and randomly selected data between a specific period (from 17^{th} July 2015 to 23^{rd} October 2015). Statistical validation of performance measured by precision and recall that are evaluated, recorded, and describes in the following sections.

6.3.1 Distance metric analysis for K-nearest neighbour

K-nearest neighbour (KNN) classification has been performed by measuring the distance between the test data and the training data using distance function.

Though Euclidean has been widely use in this field, it is prudent top investigate different distance function which can affect the KNN performance over the datasets. This experiment is based on four different distance functions including Euclidean, Jaccard, City-Block, Minkowski are used during KNN classification individually [93].

The testing has been done on the daily TU data with 10% training set. In addition the 'K' has been varied by 1NN, 3NN, and 5NN. The outcomes are tabulated in the Table 6.1. It has been compared by calculated the correct rate, error rate, precision and recall for both the training and the testing data.The experimental results show that for all the cases Euclidean achieved better performance among other, while Minkowski shows bad performance compared to other three distance metrics. For training most of the obtained good results but for training the performance is noticeable for further consideration. Thus, Euclidean distance has been considered for further investigation and comparison.

TABLE 6.1: Distance metric based analysis of KNN for one day TU data with 10% traning data.

Classifiers	Correct Rate		Error Rate		Recall		Precision	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
1NN-Jacard	1.000	0.547	0.000	0.452	1.000	0.866	0.818	0.666
1NN-Cityblock	1.000	0.656	0.000	0.343	0.723	0.698	1.000	0.680
1NN-Minkowski	1.000	0.522	0.000	0.477	0.839	0.769	1.000	0.673
1NN-Euclidean	**1.000**	**0.898**	**0.000**	**0.102**	**1.000**	**0.949**	**1.000**	**0.873**
3NN-Jacard	0.842	0.487	0.158	0.513	0.923	0.789	0.999	0.687
3NN-Cityblock	0.782	0.440	0.218	0.560	0.846	0.695	0.998	0.677
3NN-Minkowski	0.761	0.453	0.239	0.547	0.825	0.752	1.000	0.618
3NN-Euclidean	0.982	0.607	0.018	0.393	1.000	0.961	0.999	0.680
5NN-Jacard	0.998	0.601	0.002	0.399	1.000	0.930	1.000	0.687
5NN-Cityblock	0.999	0.616	0.001	0.384	0.857	0.633	1.000	0.680
5NN-Minkowski	0.993	0.642	0.007	0.358	0.658	0.490	0.686	0.448
5NN-Euclidean	1.000	0.655	0.000	0.445	1.000	0.897	1.000	0.663

6.3.2 MC-SVM results analysis

The obtained results from the baseline method MC-SVM have been analysed here in terms of statistical validation by precision and recall calculation are evaluated

and recorded in Table 6.2. The highest precision and recall value obtained from 10% training and 90% testing data (marked in bold) in all the cases using MC-SVM. This algorithm achieved an excellent performance with 99.3% of precision in randomly selected data, which signifies a high positive predictive value for all predicted TUs. Also observed that the performance increased when the training data volume is being increased. The highest recall 96.3% is achieved in case of randomly selected data, which directs the relevant prediction of TUs with a very high accuracy. Thus, experimental result signifies that the algorithm can make fault predictions about TU behaviour with the help of very few amount (only 10% is enough) of training data.

TABLE 6.2: Classification results by MC-SVM.

Observations		10%	20%	30%
Daily	NOB	656	595	536
	Precision	**0.983**	0.982	0.979
	Recall	**0.921**	0.913	0.897
Weekly	NOB	3262	2970	2672
	Precision	**0.979**	0.978	0.977
	Recall	**0.905**	0.902	0.896
Monthly	NOB	12944	11705	10643
	Precision	**0.988**	0.988	0.987
	Recall	**0.943**	0.942	0.938
Randomly	NOB	29744	27002	24339
	Precision	**0.993**	0.992	0.992
	Recall	**0.965**	0.964	0.964

6.3.2.1 Confusion matrix analysis

Confusion matrices are set to concentrate on the type of TUs that are assigned in wrong category by the algorithm. Four confusion matrices are shown in Figure 6.4. The matrices are prepared based on the highest accuracy obtained by MC-SVM (where only 10% training data are used). Each element in this matrix is the number of test items with true class in row wise and predicted class in column wise. The correctly classified objects are plotted diagonally (marked in blue color)

and misclassified objects are marked in light blue. Though the overall performance of the proposed algorithm is good, still some characteristics of TUs confound the classifier which causes misidentification. It is clearly seen that the objects of class-1 are misidentified mostly in case of daily (Figure 6.4(a)) and class-2 in case of weekly (Figure 6.4(b)) analysis. In Figure 6.4(a) and 6.4(b), 6 and (11+7+3+17)=38 objects are misidentified because, the TUs belong to these classes are distinct by the cooling and heating temperature but similar in nature with respect to power. In case of monthly and randomly (Figure 6.4(c) and 6.4(d)) data, misclassifications occurred mostly from class-0, whereas this class holds the good behaving and distinct TUs with respect to temperature and power consumption. A huge number of objects belong to this class and numerically the differences between the objects are very small. Therefore, the classifier is over fitted and overreacted to the slight fluctuations in the feature values.

6.3.3 Comparison results

Further this experimental result is compared with another two classification methods, KNN [92] and MLP [2], which are well-established method for classification. In KNN, as explained earlier it assigns the class number to an object based on the nearest distance of its neighbours. Here, K is chosen as 1 and 3 and the results are shown in Table 6.3. Table 6.3 shows the precision and recall outcomes for 1NN and 3NN with 10%, 20% and 30% training samples. Thus, it is observed from the result table that 30% training sample is required for 1NN and 3 NN to produce maximum precision and recall values, except in case of 1NN for daily analysis 20% training data is needed for high performance. Thus, it is determined that MC-SVM provides high precision and recall with less number of training data than KNN classifier.

Predicted

Actual / n=656	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5
Class 0	33	2	0	0	0	0
Class 1	0	431	0	1	0	0
Class 2	0	2	58	0	0	0
Class 3	0	1	0	51	0	0
Class 4	0	0	4	0	35	0
Class 5	0	1	0	0	0	37

(a) Daily TUs

Predicted

Actual / n=3262	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5
Class 0	176	2	11	1	1	1
Class 1	0	221	7	0	0	1
Class 2	1	2	1929	0	2	0
Class 3	0	0	0	220	0	7
Class 4	2	1	3	0	348	0
Class 5	0	0	17	0	8	301

(b) Weekly TUs

Predicted

Actual / n=12944	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5
Class 0	7550	6	2	7	8	21
Class 1	9	1279	12	3	3	0
Class 2	26	2	811	6	0	0
Class 3	6	1	1	838	2	1
Class 4	4	1	1	2	1079	0
Class 5	6	4	8	7	3	1235

(c) Monthly TUs

Predicted

Actual / n=29744	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5
Class 0	14364	0	13	0	5	4
Class 1	0	3252	2	11	4	2
Class 2	23	2	2139	5	1	0
Class 3	46	8	0	4311	13	0
Class 4	24	0	2	11	3188	0
Class 5	21	1	6	2	6	2278

(d) Randomly Selected TUs

FIGURE 6.4: Confusion matrix by MC-SVM.

TABLE 6.3: Precision and recall comparison of 1NN and 3NN.

Observations		10%		20%		30%	
		1NN	3NN	1NN	3NN	1NN	3NN
Daily	NOB	654	656	588	593	525	533
	Precision	0.945	0.916	**0.957**	0.944	0.949	**0.961**
	Recall	0.774	0.686	**0.818**	0.772	0.786	**0.829**
Weekly	NOB	3265	3274	2966	2964	2680	2697
	Precision	0.957	0.950	0.969	0.967	**0.972**	**0.973**
	Recall	0.818	0.792	0.863	0.855	**0.877**	**0.877**
Monthly	NOB	12942	12968	11693	11721	10600	10621
	Precision	0.968	0.967	0.971	0.975	**0.974**	**0.976**
	Recall	0.861	0.855	0.871	0.886	**0.885**	**0.890**
Randomly	NOB	29758	29766	26912	26896	24411	24383
	Precision	0.969	0.970	0.974	0.976	**0.979**	**0.977**
	Recall	0.865	0.865	0.882	0.891	**0.904**	**0.898**

Conversely, Table 6.4 shows the precision and recall outcomes for MLP with 10%, 20% and 30% training and 90%, 80%, and 70% samples. It is observed from the result table that 20% training sample is required to produce maximum precision and recall values for daily, weekly, monthly, and for random analysis for high performance. Thus, it is determined that MC-SVM provides high precision and recall with less number of training data than MLP neural network.

TABLE 6.4: Classification results by MLP.

Observations		10%	20%	30%
Daily	NOB	656	595	536
	Precision	0.847	**0.962**	0.825
	Recall	0.411	**0.520**	0.490
Weekly	NOB	3262	2970	2672
	Precision	0.953	0.934	**0.957**
	Recall	0.350	0.320	**0.375**
Monthly	NOB	12944	11705	10643
	Precision	0.876	**0.854**	0.813
	Recall	0.498	**0.530**	0.531
Randomly	NOB	29744	27002	24339
	Precision	0.866	**0.883**	0.798
	Recall	0.562	**0.633**	0.648

The precision and recall values are plotted in Figure 6.5 to compare the performance of MC-SVM, 1NN, 3NN, and MLP for 10%, 20%, and 30% training set.

Precision and recall are shown in blue and yellow colour respectively. It is observed that MC-SVM achieved higher precision and recall than 1NN, 3NN, and MLP for all time spans. Therefore, the obtained accuracy of the MC-SVM exhibit the effectiveness of this algorithm to learn and predict for faulty and non-faulty HVAC TUs, which is approximately 99% precision and 96% recall. The above fault detection and diagnosis results using the MC-SVM algorithm indicates, this method is robust and effective to recognize any types of faulty and non-faulty TU behaviours.

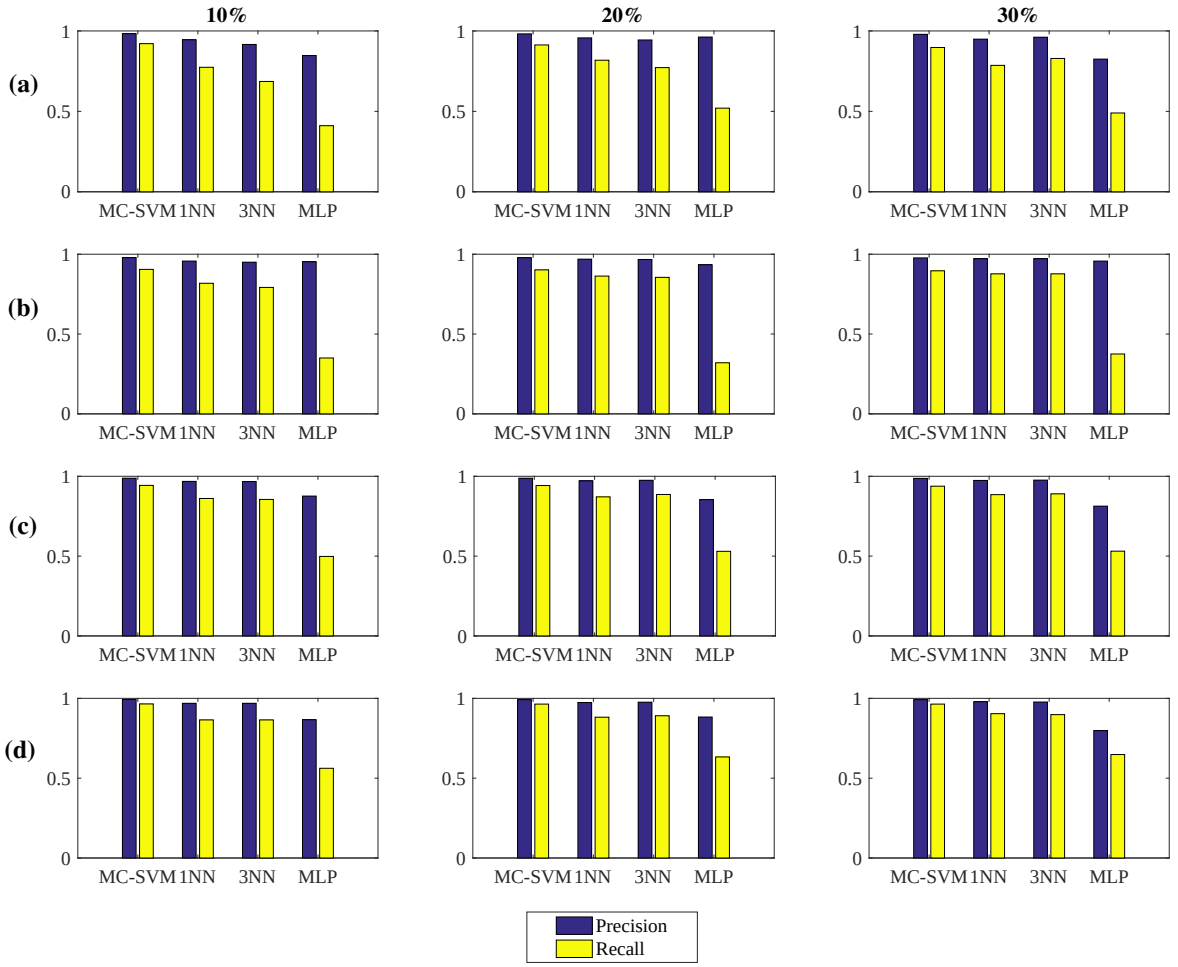


FIGURE 6.5: Precision and recall value analysis for TU data of (a) Daily, (b) Weekly, (c) Monthly, and (d) Randomly.

6.4 Conclusion

This chapter concludes on the classification methods which was applied to predict and diagnose the faults within a building automatically. The theory behind the models and their working principles have been presented. The result analysis based on the different obtained algorithms have been explained. Although, it is observed that MC-SVM performed well in identifying the different faulty and non-faulty TU behaviours among the compared methods. Therefore, suitability of MC-SVM for this building's TU data is evaluated by statistical measures and considered as baseline classification method for these terminal unit behaviour research. Due to the more realistic implementation i.e. lack of knowledge of sensor location, weather data, and the occupancy information, it is difficult to detect and diagnose the cause of faults effectively here, though the excellent accuracy of the proposed method signifies the real world effectiveness of the work.

This experiment has been tested on approximately forty thousand TUs and gathered the ground truth information. Augmenting this work includes further tests on TUs where faults such as, improper dead bands and varied set point temperatures, etc. have been identified using AFDD and rectified. This has been used to train the model for forecasting, which can predict faults on more recent data in real time by using semi-supervised learning techniques are explained in the next chapter.

Chapter 7

Semi-supervised learning techniques

"Purity, patience, and perseverance are the three essentials to success and above all, love."

-Swami Vivekananda

7.1 Introduction

The terminal units (TU) are responsible for maintaining the temperature inside a room via a constant air-flow. The quality of this air flow is being monitored using intelligent methods based on the provided historical or old dataset. The identification of the TU behaviours whether it is faulty or not has already been explored in Chapter 6 using supervised learning and successfully identified the distinct TU behaviours. In this case, these datasets have been priory labelled to classify them into 'faulty' or 'non-faulty' categories. This brings a new perspective

towards automatic fault detection and diagnosis (AFDD) TU behaviour analysis. However, if the classification needs to be done in real-time, it means all the data must be labelled prior to performing the classification continuously. This makes the 'fault' identification or classification task time complex as well as computationally heavy. It affects the whole fault finding process and it will take more time to inform the building engineer about the faults resulting in poor behaviour and energy consequences. To accommodate this obstacle, semi-supervised learning must be considered. This chapter demonstrates and evaluates semi-supervised learning (SSL) techniques for TU behaviour classification.

The rapid and dynamic growth of the data brings new challenges to BMS modelling. It is essential to use effective ML techniques rather than traditionally applied learning methods. This could help forecast different unit's behaviour more accurately and quickly. Semi-supervised learning is a class of machine learning technique's that make use of a small amount of labelled data for training with a large amount of unlabelled data for testing. It falls between unsupervised learning (without any labelled training data) and supervised learning (with completely labelled training data) methods. To ensure improved performance of automated methods promoting machine-learning techniques, a building's raw sensor data requires labelling, which increases the overall operational costs of the system employed and makes real time application difficult. Due to the limited availability of labelled information and to make the AFDD worth implementing in real-time, data driven semi-supervised learning based robust AFDD method is proposed here. Further, this method has been tested and compared for more than twenty million data points. The results presented in this chapter have a major contribution in obtaining the established statistical performance metrics and paired t-test have been applied to validate the proposed method.

7.2 Proposed semi-supervised learning method

Semi-supervised learning models have gained popularity than the traditional learning methods of fully supervised or unsupervised. Semi-supervised learning exploit both labeled and unlabeled data and have shown good performance in many applications [96]. In real-world situations, most of the data are unlabeled and need to be fit for particular purpose by labelling a small amount of dataset for unlabelled learning. The limited availability of appropriate 'fault' information, enormous levels of data for pre-processing and labelling before testing can be executed. This of course is time consuming but common with real world scenarios where data is not always in appropriate formats. Previously proposed unsupervised and supervised machine learning algorithms (described in Chapter 5 and Chapter 6) were investigated to create the lablled data set for training from TUs over a given period from the same buildings under test for the experiments [97].

In this chapter, SSL is introduced to develop a framework that would identify the 'faulty' and 'non-faulty' behaviour to predict their behaviour using highly confident unlabeled data. Therefore, SSL based multi-class support vector machine (MC-SVM) is employed for AFDD and established through training, testing, and validation process. The proposed AFDD methodology consist of several stages and the step-wise procedure of the proposed architecture is shown in Figure 7.1.

These feature extraction steps (as shown in Figure 7.1) intend to derive informative and non-redundant values about TU characteristics, which helps the proposed semi-supervised learning framework in the identification of significant TU patterns. In this test, six different classes of faulty and non-faulty TU patterns are available for specific period and used as labelled data. Then, multi class support vector machine (MC-SVM) is employed into SSL framework for classifying the faulty and non-faulty TU patterns [98]. This SSL model is simple yet more efficient and adopts three steps: training, testing and validation. Subsequently, the training

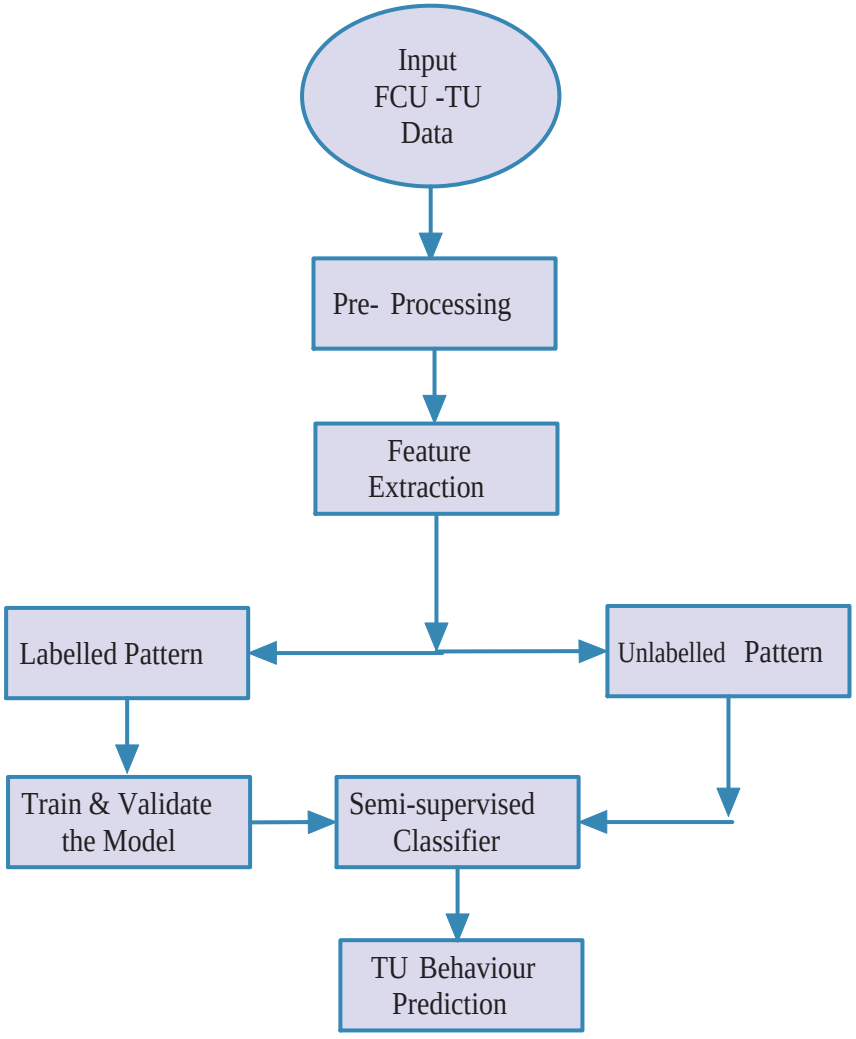


FIGURE 7.1: Proposed AFDD architecture for semi-supervised learning approaches.

and testing accuracy of the proposed model have been measured through precision and recall calculation. Thereafter, new unlabelled data are fed into the best scored SSL model to predict the faulty and non-faulty TU patterns. This prediction then validated through paired t-test [99, 100] which has been determined for understanding the correlation between historical data (labelled) and predicted data (unlabelled).

7.2.1 Multi-class support vector machine and kernel function

Multi-class support vector machine (MC-SVM) has been used to perform the training and testing task in this semi-supervised learning approach. The algorithm of MC-SVM and it's working principles have already been discussed in the previous chapter, where it has been employed for supervised approach. It has been defined that six classes are present in this TU behaviour classification. Thus, MC-SVM has been employed rather than the general SVM as this is suitable for binary classification. Here, a group of mathematical functions are determined as kernel function, which are applied for choosing the decision boundary margin for MC-SVM classifier. Kernel function take the input data and convert it to the desired form in a implicit feature space with view to make it linear dataset. Different kernels are simply different in case of making the hyper plane decision boundary between the classes. Depending upon the application and data characteristics different types of kernel functions has been used [101, 102] such as, linear, polynomial, radial basis function (RBF), etc. Here, two type of kernel functions have been used for the experiments: liner kernel and quadratic kernel. In addition, quadratic kernel is a polynomial kernel, it uses the degree of the parameter (p) as 2, thus it known as quadratic. The linear kernel and quadratic kernel functions are defined as,

$$k(x_i, x_j) = (x_i \ . \ x_j) \tag{7.1}$$

$$k(x_i, x_j) = ((x_i \ . \ x_j) + 1)^p \tag{7.2}$$

Where, k is the kernel functions and $(x_i \ . \ x_j)$ is the dot product of the two feature vector, p is the adjustable parameter (here p =2) and requires one addition and exponentiation on the original dot product.

7.2.2 Model validation

Precision and recall (described in Section 6.2.4) have been measured to validate the training and testing phase where label information are available which assist to find out the true TU predictions (truly faulty and non-faulty TU) and false predictions (wrong TU class prediction). Precision and recall are then calculated from these true and false predictions. SSL has been applied to the unlabelled data therefore true and false predictions could not be calculated as before. Thus, the paired t-test has been investigated to estimate the correlation between a labelled class and the same TU class predicted by the SSL algorithm. Therefore, the null hypothesis symbolise the fitness of a predicted TU class data with the TU belongs to that class in historical data. Test result delivers one to denote the rejection of consideration of predicted data in the same class of labelled data and zero for acceptance based on the probability (p-value) of test observation. Low probability or p-value implies the invalidity of null hypothesis. The null hypothesis (H_0) assumes that the true mean difference (μ_d) is equal to zero as shown below,

$$H_0 : \mu_d = 0 \tag{7.3}$$

The results of precision, recall, and t-test have been discussed in the result analysis below.

7.3 Result analysis

The faulty and non-faulty TUs classification for the new unlabelled data has been performed here using the historical labelling data, and have already been performed previously (in Chapter 5). The novel feature extraction method (proposed

earlier in Chapter 4) was employed on the new multi-stream TU dataset for dimensionality reduction. Thereafter the featured data used for the data-driven TU fault detection and diagnosis purpose employing semi-supervised multi-class support vector machine.

7.3.1 Experimental data details

This experiment has been tested and results are observed for two commercial building in London over a period from July 2015 to July 2018. A single TU consists of multiple data streams; such as, control temperature, set point, dead band, heating and cooling power, enable signals (for this test more than 50 million TU data points are considered). An example of signal TU data for January 2016 of case study-1 is shown in Figure 7.2. Here the blue lined graph denotes control temperature variation with respect to the heating and cooling set point and corresponding power demand (shown in red) for a month during winter from a building based in central London. The x-axis shows number of days and y-axis shows temperature in the below graph and corresponding power demands on the other graph.

This investigation has been performed on two case study buildings detailed in the Section 3.3. First the building of case study-1 comprises 17 floors and 723 operating TUs all spread across different floors. The experiments have been executed on the second building of case study-2 which has altogether 490 operating TUs across 7^{th} floors. This experiment was conducted using data covering a period of three years from these two case-studies. The data labelling procedure has already being explained in the previous chapters. Labelled data have been used to train the model and test on different days. This study have been started for the 17^{th} of July 2015 for case study 1 and the 1^{st} of January 2016 for case study 2. These whole period of data has been considered to train the model with the help of labelled information. This involves both training and testing with varying percentages to

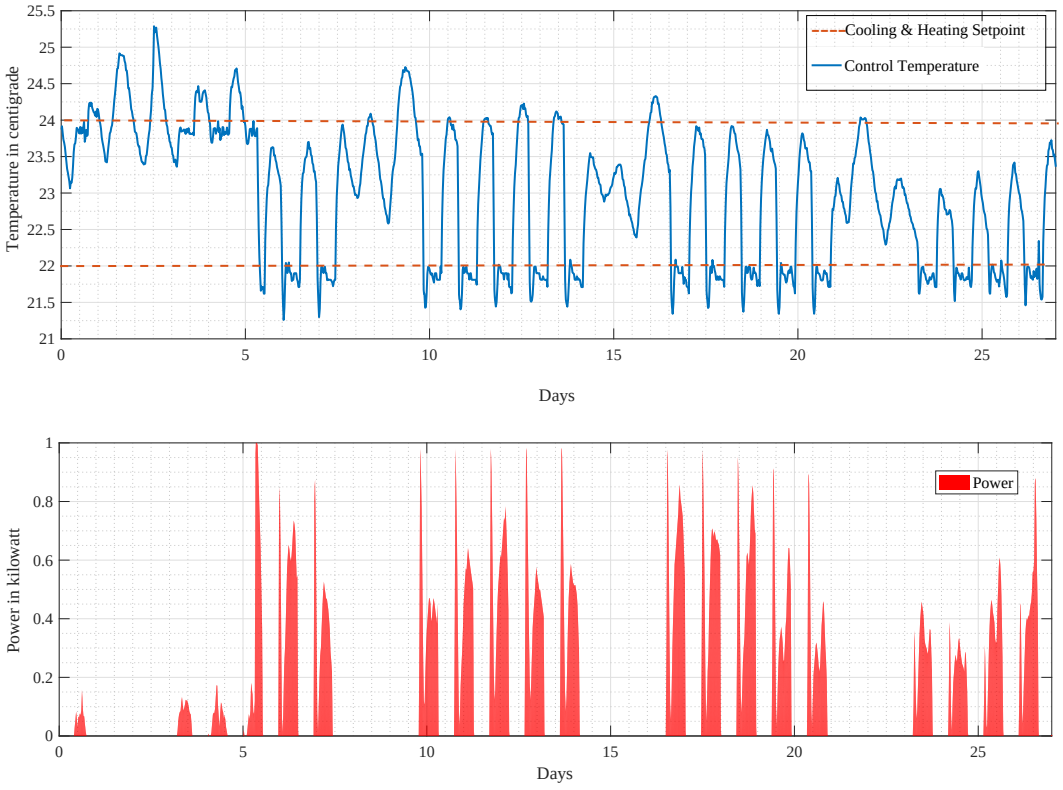


FIGURE 7.2: An example of a TU control temperature and power data signal over 30 days in winter.

understand how well the data will train with the dataset. Then, TU behavioural prediction has been performed by SSL where a label is unavailable. This training and testing have been done as a trial basis for two seasons mainly: summer and winter over a period of 3 years. The first year data has been considered for training the model and rest of the data used for testing purpose. In addition it has been kept in mind that the training data should have the varieties of different seasons (mainly summer and winter) so that it can classify different seasons data desirably. The experimental data for both the case studies are tabulated in Table 7.1. In the table labelled data have been used for training the model and unlabelled for testing the model.

TABLE 7.1: Obtained data details for the SSL experiments.

Data Details	Case Study 1		Case Study 2		Seasons
	From	To	From	To	
Labelled	17th July 2015	23rd October 2015	1st January 2016	31st July 2016	Summer & Winter
Unlablelled	1st November 2015	31st July 2018	1st August 2016	31st July 2018	Summer & Winter

7.3.2 Results analysis of case study-1

The SSL model has been trained using different training data and investigated by two classification algorithms with variable parameter tuning. The data set have six classes including three types of faulty and three types of non-faulty TU patterns (described in Chapter 5). The baseline classification method multi-class support vector machine (MC-SVM) has been used and K-nearest neighbour (KNN) [85] has been performed for model comparison. In the case of the KNN experiment, the 'K' has been varied by one, three, and five while MC-SVM has been experimented using two kernel functions, linear (LMC-SVM or LSVM) and quadratic (QMC-SVM or QSVM). The obtained testing accuracy results are compared and tabulated in Table 7.2.

TABLE 7.2: Testing accuracy results obtained by different methods for case study 1.

Methods	10%		20%		30%		40%		50%		60%	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re
1NN	0.692	0.744	0.774	0.902	0.802	0.918	0.734	0.812	0.774	0.887	0.788	0.902
3NN	0.772	0.987	0.798	0.955	0.789	0.939	0.786	0.891	0.809	0.932	0.823	0.934
5NN	0.799	0.976	0.813	0.961	0.829	0.967	0.804	0.955	0.811	0.919	0.815	0.931
LSVM	0.771	0.995	0.785	0.987	0.831	0.973	**0.837**	**0.978**	0.821	0.946	0.824	0.981
QSVM	0.691	0.898	0.745	0.91	0.753	0.919	0.813	0.948	0.741	0.914	0.772	0.892

The experiment has been executed by selecting training and testing data randomly from each of the specific training and testing set (as given in Table 7.2). The training data have been varied from 10% to 60% and vice versa for testing purpose. The training and testing performances have been calculated separately to check the robustness of the proposed model. The highest recall (0.987) has

achieved in 3NN using 10% of training data and highest precision (0.823) in 3NN using 60% training data. KNNs have performed competitive with MC-SVMs in training phase because KNNs find the distance between data points in feature space and a nearest neighbour is the data point itself in the training period, where MC-SVMs find the inner product or solve quadratic function to discover the best margin among support vectors that do not deliver as good a result as KNN. 1NN has worked well because of data compactness. In the case of testing phase, five nearest neighbour votes deliver improved TU pattern recognition than 1NN. Conversely, LMC-SVM has gained the highest recall (0.978) in the testing cases for 40% training data variations. In addition, LMC-SVM has achieved better testing precision (0.837 with 40% training data). The bar graph representation of the obtained performance in testing phases using different classification algorithms has been compared and shown in Figure 7.3. In terms of overall precision and recall, linear kernel has worked better than the other algorithms. The linear kernel function defines the optimum margin in feature space. Therefore, LMC-SVM has obtained highest performance score among other classifiers. Thus, LMC-SVM model with 40% training data has been considered most efficient predictor for this TU data using the SSL approach.

7.3.3 Results analysis of case study-2

The same experiment have been performed on the building of case study-2 to check the robustness of the proposed model. The SSL model has been trained by data over a period as shown in Table 7.1. The training has been performed on labeled data. Therefore, the data labeling has been done by employing X-means clustering as described in the previous chapter (Chapter 5). The six different faulty and non-faulty classes have been identified. Thereafter the SSL approaches have been performed to illustrate the effectiveness of the proposed approach. This classification is performed by KNN [85] and MC-SVM for the classification and

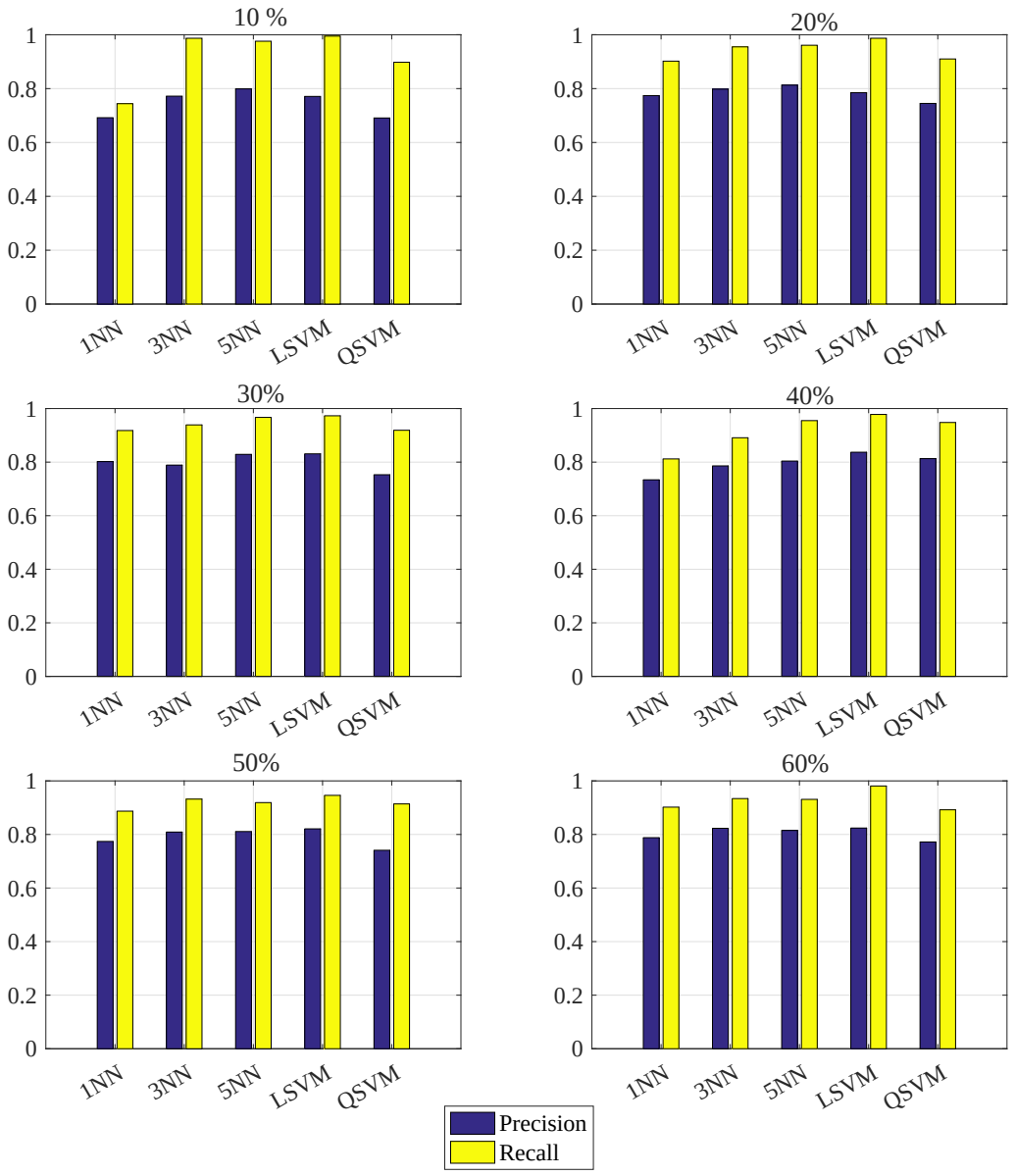


FIGURE 7.3: Comparison of precision and recall score for case study-1.

method comparison. For KNN, the 'K' has been varied by one, three, and five. For MC-SVM, two kernel function linear (LMC-SVM) and quadratic (QMC-SVM) has been used. The obtained testing accuracy based on the training variation have been tabulated and compared in Table 7.3. The experiment was also conducted by randomly dividing the data set into training and testing phases. The training data have been varied from 10% to 60% and vice versa for testing. Both the

training and testing performances have been calculated separately to investigate the robustness of the proposed model and the testing results are shown here. It can be observed based on the KNN classification outcomes that the highest precision (0.816) and recall (0.928) have been achieved in 1NN using 30% of training data. But, for 3NN and 5NN highest precision (0.825) and (0.830) have been achieved with 30% and 60% training data. Highest recall (0.986 and 0.955) for both of them achieved with 10% training data. So, overall it can be remark that 1NN with 30% training data will be good for classification in terms of both precision and recall outcomes.

On the other hand, LMC-SVM has also gained the high recall values for all the different amount of data variations. In addition, LMC-SVM has achieved better testing precision (0.831) and recall (0.986) with 40% training data than the training percentage. Although, in QMC-SVM best results were obtained with 40% training data where precision and recall are 0.825 and 0.958 respectively. Form this experiments it has been found that in terms of overall precision and recall, linear kernel (LMC-SVM) has performed superior than other algorithms for both the case studies. Therefore, it is assumed that the linear kernel function defines the optimum margin in the feature space of these TU data. The LMC-SVM has found appropriate for this TU data classification and further the hypothesis validation has been performed for this classifiers. The bar graph of the outcome table for the testing phases have been shown in Figure 7.4.

TABLE 7.3: Testing accuracy results obtained by different methods for case study-2.

Methods	10%		20%		30%		40%		50%		60%	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re
1NN	0.683	0.754	0.784	0.912	**0.816**	**0.928**	0.754	0.810	0.754	0.889	0.786	0.913
3NN	0.775	0.986	0.778	0.935	0.782	0.969	0.726	0.901	0.813	0.922	0.825	0.914
5NN	0.789	0.955	0.821	0.952	0.830	0.927	0.805	0.952	0.808	0.911	0.826	0.932
LSVM	0.731	0.965	0.795	0.975	0.829	0.976	**0.831**	**0.986**	0.813	0.957	0.821	0.979
QSVM	0.771	0.858	0.732	0.920	0.761	0.920	0.825	0.958	0.732	0.944	0.778	0.873

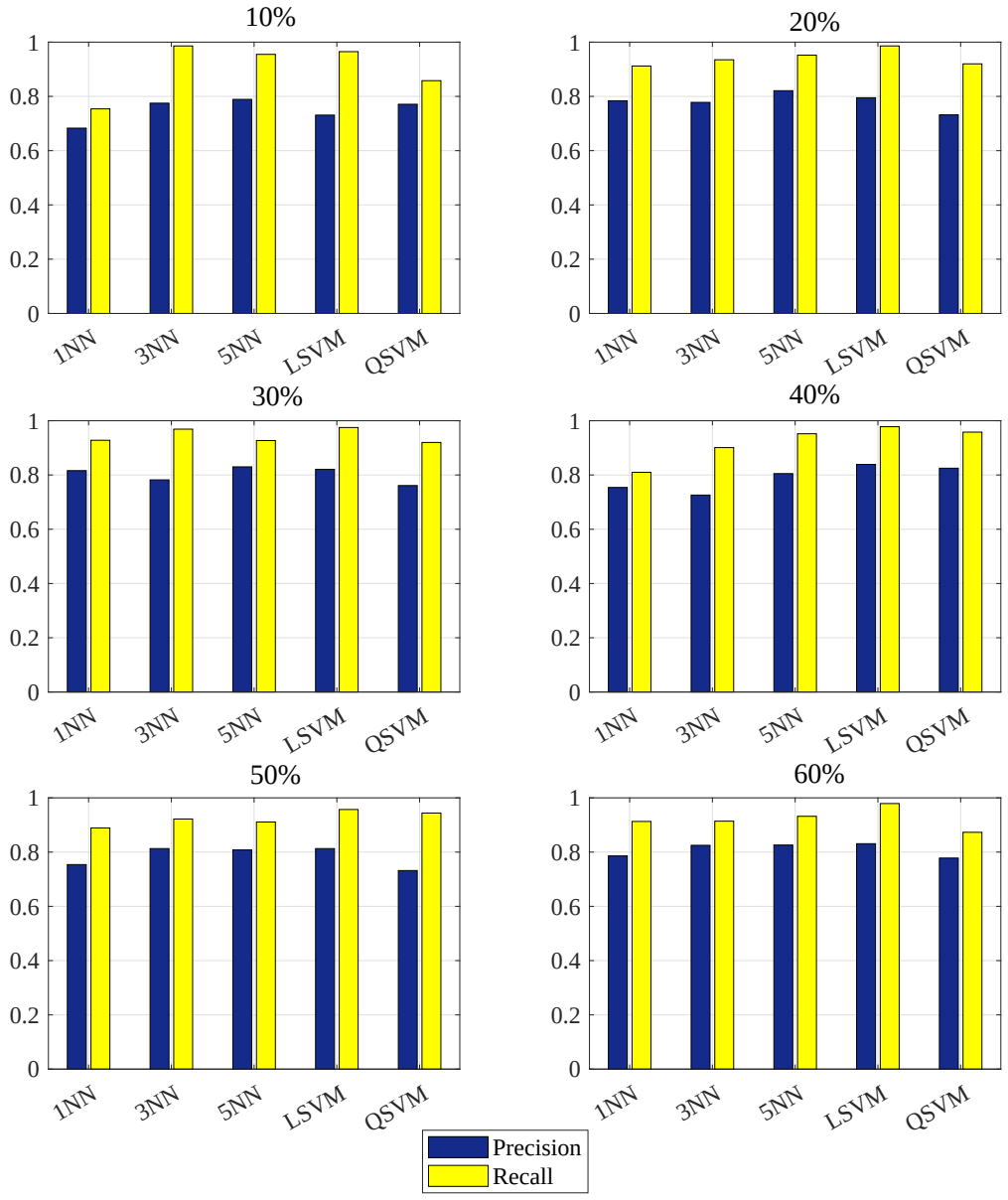


FIGURE 7.4: Comparison of precision and recall score for case study-2.

7.3.4 Validation on unlabelled data

In case of the case study 1, LMC-SVM model with 40% training data has been considered the most efficient predictor for the SSL approach. Further, this model is used here for the TU prediction without label information. Consequently, paired t-test has been implemented to discover the correlation between the predicted TU

class and the TUs truly belong to that class. Figure 7.5 shows the comparison of p-values obtained by the semi-supervised LMC-SVM for different TU classes where, the first three classes represent the non-faulty TU patterns in terms of control temperature and corresponding power demands. Other three classes represent the different faulty TU patterns. The significance level 0.05 has been considered and the p-value has been determined for a TU class to justify the null hypothesis. The null hypothesis has been accepted for a predicted class where p-value is more than the significance level. Figure 7.5 shows that the predicted class-1 and class-6 have failed to fit in the actual classes, i.e. the semi-supervised LMC-SVM could predict the class-2, 3, 4, and 5 correctly but unsuccessful in predicting the TUs from class-1 and class-6. It is observed from the results that available training data for class 1 and 6 might not be sufficient to train the LMC-SVM.

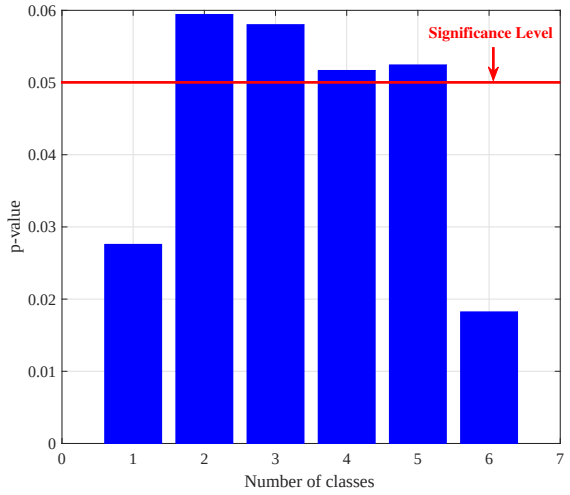


FIGURE 7.5: Obtained p-values for different classes from SSL approach using LMC-SVM for case study-1.

In the case study 2, also as the 40% training data has been found to show the highest precision and recall for the LMC-SVM model. Thus, it has also been considered the most efficient predictor for the SSL approach for this building's TU data. Perhaps, this model is used here for the TU prediction without using labelled data. Consequently, paired t-test has been used to find out the correlation between the predicted and actual class. Figure 7.5 shows the comparison of p-values obtained by the semi-supervised LMC-SVM for this case study-2 building's

TU. As, the significance level is 0.05 and the p-value must be above the significance level for each class to accept the null hypothesis of the outcomes. Figure 7.6 shows the p-values and found that it is higher than the case study-1 but still failed to fit in the actual classes for class-1 and class-6. Semi-supervised LMC-SVM is able to predict the class-2, 3, 4, and 5 correctly but unsuccessful in predicting the TUs from class-1 and class-6. Further investigation is underway to overcome this limitation with more training data which can deal with these class 1 and 6 as might the amount of data in these two classes are inadequate to train the LMC-SVM model properly.

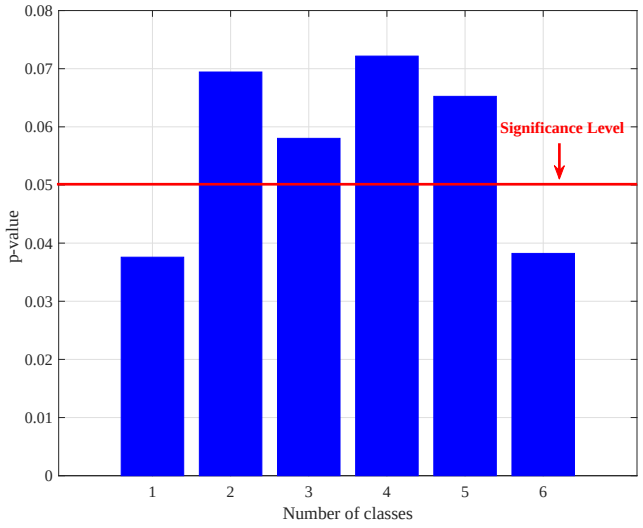


FIGURE 7.6: Obtained p-values for different classes from SSL approach using LMC-SVM for case study-2.

7.4 Conclusion

It is concluded from the presented research outcomes that the semi-supervised learning (SSL) approach for automatic fault detection and diagnosis study is challenging and worth investigating. Here, faulty and non-faulty TU prediction has been investigated using two different classification algorithms varying five different parameters. The results show that it performs well (overall precision and recall are above 0.8 and 0.95) when compared to the different KNN algorithm for both case

studied building's TU data. New and unlabelled data can be effectively classified using this approach depending upon the previously available historic dataset, which have been labelled priory employing Xmeans clustering. It is also found that the performance of LMC-SVM is the best-fitted model among five-tested methods. Based on the paired t-test results, LMC-SVM would need to be improved in the one faulty and one non-faulty class types. Thus, more training data could effectively solve these issue and other classification algorithms investigated to improve future SSL performance.

Chapter 8

Conclusions of the thesis

.

"Take risks in your life. If you win, you can lead, if you lose, you can guide."

-Swami Vivekananda

8.1 Introduction

Computerized building management system (BMS) provides an opportunity to facilitate building maintenance and render optimal operational performance. The foremost concern in the building sector is to optimise the excess energy consumption and maintain the performance inside buildings. One of the major units used in buildings is the heating, ventilation, and air-conditioning (HVAC) system which consumes most of the effective energy and is the primary unit for maintaining user comfort in a building. Specifically, this research has focused on small HVAC units i.e., terminal unit (TU) of FCU. Moreover, BMS plays a vital role when TU work

inefficiently as they are vast in numbers. However, it is challenging to perform analytics on enormous and amorphous BMS databases. Additionally, finding faults through manual searching is exhaustive, requires expert views, time consuming, expensive, and this is why so many energy problems are left undetected/ignored. Currently, systems cannot autonomously predict future scenarios to anticipate faults.

8.2 Remark on the proposed automatic fault detection and diagnosis system

It has been concluded from the research outcomes that the proposed automatic fault detection and diagnosis (AFDD) model and the technical machine learning sequence is the best possible solution for buildings to predict their behaviour and faults anticipation. Initial research started by developing a system to demonstrate how predictive control can be influential in a commercial building for faulty heating and cooling TU units. This study has described the benefits of the proposed approach to identify the faults remotely in building and forecasting the behaviour automatically to notify building engineer for taking necessary actions. The proposed strategy has been built based on unsupervised, supervised and semi-supervised learning techniques. These learning enable building to be more intelligent through automatic monitoring the heating and cooling temperature inside building. In consequence, these systems would be able to predict future situations and anticipate faults to optimize building operation and their energy use.

Real TU behaviour is analysed remotely in this work, where temperature set point and corresponding power consumption are considered as the parameters for behaviour characterisation. These parameters enable a story to form around the TU and its direct environment, i.e. providing real data on the building under test

and enabling managers to investigate faulty TUs. Three distinct fault and three non-fault clusters have been remotely identified and classified. From the AFDD results, it can be concluded that faulty behaviour can be automatically and remotely detected and understood. Subsequently useful and exact information can be delivered to those managing the TUs in question. Using this vital new information an investigator can then identify if it is the environment (set-point issues) or the particular TU creating the fault. The predictive fault finding can be achieved using the classification results presented through SSL, thus managers can ensure timely interventions. In some cases where a fault is assumed, but actual fact is the TUs in question once checked are not "broken" but for example have had their temperature set points set too high or low for their environmental setting, making it difficult for those TUs to achieve set point temperature leading to "faulty" behaviour displays. Additionally, too narrow dead band settings (difference between high and low set points) ranges cause many potential faulty behaviour. Also, external influences such as weather conditions, sunlight, photocopiers, fridges, manual set point changes (based on personal preferences), open windows, etc. are not considered for both the set point and dead band levels estimation. Usually set point changes in situations above can be triggered by the building occupants depending upon their desired comfort levels and the effect of such request on TU behaviour is observable. Affected TUs subsequently demand excess power but will be unable to maintain the control strategies requested. Importantly the TUs in this case are not always defective but simply trying to achieve unattainable goals. Thus, energy and OPEX savings can be quickly made by addressing these external issues and the solution satisfies the result was driven at the end of thesis as having major contribution in the provides this assistance for the proposed AFDD system.

8.3 Future work

This research will concentrate guiding the previous AFDD study to building maintenance services by incorporating text-mining method. This will explore the further use of the system with different buildings data to improve the performance and provide guidance for building operators. This can be done by analysing and tracking the error messages provided by building engineers and service providers. Investigating and learning the text of the message and combining with the developed AFDD system could make the system more intelligent and further contribute towards making the building smarter informing best practice. These could lead to improve BMS control strategies, identify faulty and failing equipment that is misbehaving and making excessive demands on the plant rigorously. The method will be executed in the following steps:

1. Step 1- Historic text data collection, provides the received messages for the behaviour of HVAC units and respective suggestions and completed actions.

2. Step 2- Pre-processing of the text data to convert work order descriptions into a mathematical format that provides itself to a quantitative lexical analysis.

3. Step 3- Clustering to focus on interesting sections of database that contain information about faults in building systems and components – rather than less interesting routine maintenance and inspection activities.

4. Step 4 Association rule-mining with the fault classification from the developed AFDD to identify the coexistence tendencies among the cluster of interest.

The power and flexibility of a modern BMS allows energy control strategy to be implemented and the changes in user requirements, developments in HVAC

technology and evolving policies on energy conservation ensure that innovation and advancement in building design is a continuing process.

Publications

Journals

1. S. P. Rana, M. Dey, G. Tiberi, L. Sani, A. Vispa, G. Raspa, M. Duranti, M. Ghavami, S. Dudley. "Machine Learning Approaches for Automated Lesion Detection in Microwave Breast Imaging Clinical Data", Nature Scientific Reports 9 (1), 10510. (Impact factor- 4.525).

2. S. P. Rana, M. Dey, M. Ghavami, S. Dudley. "Non-Contact Human Gait Identification through IR-UWB Edge Based Monitoring Sensor." IEEE Sensors Journal (2019), DOI: 10.1109/JSEN.2019.2926238. (Impact factor- 3.076).

3. M. Dey, S. P. Rana, P. Siarry. "A robust FLIR target detection employing an auto-convergent pulse coupled neural network." Remote Sensing Letter (2019), DOI: 10.1080/2150704X.2019.1597296. (Impact factor- 2.024).

4. S. P. Rana, M. Dey, M. Ghavami, S. Dudley, "Signature Inspired Home Environments Monitoring System Using IR-UWB Technology" Sensors, 19(2). (Impact factor- 3.031).

5. S. P. Rana, M Dey, P. Siarry, "Boosting Content Based Image Retrieval Performance Through Integration of Parametric & Nonparametric Approaches", Journal of Visual Communication and Image Representation, 58 (2019): 205-219. (Impact factor- 2.259).

6. S. P. Rana, J. Prieto, M. Dey, S. Dudley, J. Corchado, (2018). "A Self Regulating and Crowdsourced Indoor Positioning System through Wi-Fi Fingerprinting for Multi Storey Building", Sensors, 18(11), 3766. (Impact factor- 3.031).

7. M. Dey, S. P. Rana, S. Dudley, "Smart Building Creation in Large Scale HVAC Environments through Automated Fault Detection and Diagnosis", Future Generation Computer System, 2018. (Impact factor- 5.768).

Conferences

1. S. P. Rana, M. Dey, M. Ghavami, S. Dudley, "ITERATOR: A 3D Gait Identification from IR-UWB Technology", The IEEE 41st International Engineering in Medicine and Biology Conference (EMBC), Berlin, Germany, July 23–27, 2019.

2. S. P. Rana, M. Dey, M. Ghavami, S. Dudley, "Intelligent Three-dimensional Gait Analysis Using IR-UWB Sensing", The 41st Progress in Electromagnetics Research Symposium (PIERS), Rome, Italy, June 2019.

3. M. Dey, S. P. Rana, S. Dudley, "Semi-Supervised Learning Techniques for Automated Fault Detection and Diagnosis of HVAC Systems", The IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), Volos, Greece, November 2018.

4. S. P. Rana, M. Dey, R. Brown, H. Siddiqui, S. Dudley, "Remote Vital Sign Recognition Through Machine Learning Augmented UWB", 12th IEEE International Conference on European Conference on Antennas and Propagation (EuCAP), London, UK, April 2018.

5. M. Dey, M. Gupta, S. P. Rana, M. Turkey, S. Dudley, "A PID Inspired Feature Extraction Method for HVAC Terminal Units", 5th IEEE International Conference on Technologies for Sustainability (SusTech), Phoenix, AZ, November, 2017.

6. S. P. Rana, M. Dey, H. Siddiqui, G. Tiberi, M. Ghavami, S. Dudley, "UWB localization employing supervised learning method", IEEE International Conference on Ubiquitous Wireless Broadband ICUWB2017, Salamanca, Spain, 2017.

7. M. Dey, M. Gupta, M. Turkey, and S. Dudley, "Unsupervised Learning Techniques for HVAC Terminal Unit Behaviour Analysis ", Proceedings of IEEE International Conference on Smart City Innovations, San Francisco, CA, August 2017.

Appendix

```
from datetime import datetime, timedelta
from pandas import DataFrame
import matplotlib.pyplot as plt
import numpy as np
from pytz import utc
from hold import hold
from scipy.stats import pearsonr
from scipy.integrate import simps
from IPython import get_ipython

from dllab.utils.model import BEMSObject, do_setup,BSCollection
from dlutils.trace_exceptions import TraceNoDefinitionError

from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score

do_setup()
# Dashwood House
collection = BSCollection.load('31c5da2e-f5ed-11e3-b056-bc764e08a231')

start = utc.localize(datetime(2015, 7, 20))
end = start + timedelta(hours=24)
freq = '10Min'

i=0
areas = {}

columns = ['Label','F1', 'F2','F3','F4','F5','F6','F7','F8','F9','F10','F11','F12']
index = np.arange(731) # array of numbers for the number of samples
af = DataFrame(columns=columns, index = index)

for tu in collection.bsobjects:
```

```
print i,tu.label
i=i+1

# Creating Defifnitions

ts={}
ts['E']= tu.is_on(start,end,freq)
ts['CT']=tu.control_temp_trace(start,end,freq)

# For Heating TUs

ts['HSP']=tu.heating_setpoint_trace(start,end,freq)
ts['HP']=tu.heating(start,end,freq)
ts['IsHeating']=tu.is_heating(start,end,freq)

# For Cooling TUs

ts['CSP']=tu.cooling_setpoint_trace(start,end,freq)
ts['CP']=tu.cooling(start,end,freq)
ts['IsCooling']=tu.is_cooling(start,end,freq)
df = DataFrame(ts)

df['OutsideGoal'] = (df['CT'] < df['HSP']) | (df['CT'] > df['CSP'])
df['WithinGoal'] = df['OutsideGoal'] == 0
df['HError'] = df['OutsideGoal'] * (df['HSP'] - df['CT']).clip(lower=0)
df['CError'] = df['OutsideGoal'] * (df['CT'] - df['CSP']).clip(lower=0)

# Heating and Cooling event Start
df['HEventstart']= (df['HError']>0) * (df['CError']==0) * df['E']
df['CEventstart']= (df['CError']>0) * (df['HError']==0) *  df['E']


if (df['HEventstart'].any()==1)&(df['CEventstart'].any()==0):
    print('only heating')
    df['HEventsEnd'] = (df['HError']==0)
    # Heating Events for Temperature Response Delay - ES-RD
    df['THRD']=(((df['CT']-hold(df['CT'],df['HEventstart']))/df['CT']).abs()<0.05)*df['HEventstart'
    # Heating After Response Delay - RD-GA
    df['THARD'] =  (df['THRD'] == 0)*df['OutsideGoal']*df['HEventstart']
    # Heating goal achieved - GA-EE
    df['THGA'] = df['WithinGoal']*df['HEventsEnd']
    # Heating Events for Power Response Delay - ES-RD
    df['PHRD'] = df['HP']*df['THRD']
```

```
        # Heating After Response Delay - RD-GA
        df['PHARD'] = (df['PHRD'] == 0)*df['HP']
        # Heating goal achieved - GA-EE
        df['PHGA'] = df['WithinGoal']*df['HEventstart']*df['HP']
        # Calculating the areas
        areas['AH1'] = simps(df['THRD'])
        areas['AH2'] = simps(df['THARD'])
        areas['AH3'] = simps(df['THGA'])
        areas['AH4'] = simps(df['PHRD'])
        areas['AH5'] = simps(df['PHARD'])
        areas['AH6'] = simps(df['PHGA'])

        areas['AC1'] = 0
        areas['AC2'] = 0
        areas['AC3'] = 0
        areas['AC4'] = 0
        areas['AC5'] = 0
        areas['AC6'] = 0

        print areas['AH1']
        print areas['AH2']
        print areas['AH3']
        print areas['AH4']
        print areas['AH5']
        print areas['AH6']

        print areas['AC1']
        print areas['AC2']
        print areas['AC3']
        print areas['AC4']
        print areas['AC5']
        print areas['AC6']

    elif (df['CEventstart'].any()==1)&(df['HEventstart'].any()==0):
        print('only cooling')
        df['CEventsEnd'] = (df['CError']==0) *(df['HEventstart']==0)
        # Cooling Events for Temperature Response Delay - ES-RD
        df['TCRD'] = (((df['CT'] - hold(df['CT'], df['CEventstart'])) / df['CT']).abs() < 0.05)
        # Cooling After Response Delay - RD-GA
        df['TCARD'] =(df['TCRD'] == 0)*df['OutsideGoal']*df['CEventstart']
        # Cooling goal achieved - GA-EE
        df['TCGA'] = df['WithinGoal']*df['CEventsEnd']
        # Cooling Events for Power Response Delay - ES-RD
```

```
        df['PCRD'] = df['CP']*df['TCRD']
        # Cooling After Response Delay - RD-GA
        df['PCARD'] = (df['PCRD'] == 0)*df['CP']
         # Cooling goal achieved - GA-EE
        df['PCGA'] = df['WithinGoal']*df['CEventsEnd']*df['CP']
        #Calculating the areas
        areas['AC1'] = simps(df['TCRD'])
        areas['AC2'] = simps(df['TCARD'])
        areas['AC3'] = simps(df['TCGA'])
        areas['AC4'] = simps(df['PCRD'])
        areas['AC5'] = simps(df['PCARD'])
        areas['AC6'] = simps(df['PCGA'])

        areas['AH1'] = 0
        areas['AH2'] = 0
        areas['AH3'] = 0
        areas['AH4'] = 0
        areas['AH5'] = 0
        areas['AH6'] = 0

        print areas['AC1']
        print areas['AC2']
        print areas['AC3']
        print areas['AC4']
        print areas['AC5']
        print areas['AC6']

        print areas['AH1']
        print areas['AH2']
        print areas['AH3']
        print areas['AH4']
        print areas['AH5']
        print areas['AH6']

    elif (df['HEventstart'].any()==1)&(df['CEventstart'].any()==1):
        print('both heating and cooling')
        df['HEventsEnd'] = df['IsHeating'] *(df['HError']==0)
        df['CEventsEnd'] = df['IsCooling'] *(df['CError']==0)
        df['THRD'] = (((df['CT'] - hold(df['CT'], df['HEventstart'])) / df['CT']).abs() < 0.05) *df['HE
        # Heating Events for Temperature After Response Delay - RD-GA
        df['THARD'] =  (df['THRD'] == 0)*df['OutsideGoal']*df['HEventstart']
        # Heating goal achieved - GA-EE
        df['THGA'] = df['WithinGoal']*df['HEventsEnd']
```

```
# Heating Events for Power Response Delay - ES-RD
df['PHRD'] = df['HP']*df['THRD']
# Heating After Response Delay - RD-GA
df['PHARD'] = (df['PHRD'] == 0)*df['HP']
# Heating goal achieved - GA-EE
df['PHGA'] = df['WithinGoal']*df['HEventstart']*df['HP']
# Calculating the areas
areas['AH1'] = simps(df['THRD'])
areas['AH2'] = simps(df['THARD'])
areas['AH3'] = simps(df['THGA'])
areas['AH4'] = simps(df['PHRD'])
areas['AH5'] = simps(df['PHARD'])
areas['AH6'] = simps(df['PHGA'])
print areas['AH1']
print areas['AH2']
print areas['AH3']
print areas['AH4']
print areas['AH5']
print areas['AH6']

# Cooling Events for Temperature Response Delay - ES-RD
df['TCRD'] = (((df['CT'] - hold(df['CT'], df['CEventstart'])) / df['CT']).abs() < 0.05)
# Cooling After Response Delay - RD-GA
df['TCARD'] =(df['TCRD'] == 0)*df['OutsideGoal']*df['CEventstart']
# Cooling goal achieved - GA-EE
df['TCGA'] = df['WithinGoal']*df['CEventsEnd']
# Cooling Events for Power Response Delay - ES-RD
df['PCRD'] = df['CP']*df['TCRD']
# Cooling After Response Delay - RD-GA
df['PCARD'] = (df['PCRD'] == 0)*df['CP']
# Cooling goal achieved - GA-EE
df['PCGA'] = df['WithinGoal']*df['CEventsEnd']*df['CP']
#Calculating the areas
areas['AC1'] = simps(df['TCRD'])
areas['AC2'] = simps(df['TCARD'])
areas['AC3'] = simps(df['TCGA'])
areas['AC4'] = simps(df['PCRD'])
areas['AC5'] = simps(df['PCARD'])
areas['AC6'] = simps(df['PCGA'])
print areas['AC1']
print areas['AC2']
print areas['AC3']
print areas['AC4']
```

```
        print areas['AC5']
        print areas['AC6']

    elif (df['HEventstart'].any()==0)&(df['CEventstart'].any()==0):
        print('neither heating or cooling')

        #Calculating the areas
        areas['AH1'] = 0
        areas['AH2'] = 0
        areas['AH3'] = 0
        areas['AH4'] = 0
        areas['AH5'] = 0
        areas['AH6'] = 0

        areas['AC1'] = 0
        areas['AC2'] = 0
        areas['AC3'] = 0
        areas['AC4'] = 0
        areas['AC5'] = 0
        areas['AC6'] = 0

        print areas['AH1']
        print areas['AH2']
        print areas['AH3']
        print areas['AH4']
        print areas['AH5']
        print areas['AH6']

        print areas['AC1']
        print areas['AC2']
        print areas['AC3']
        print areas['AC4']
        print areas['AC5']
        print areas['AC6']


    print ('-----------------------------------------------------')
    af.ix[i-1]={'Label' : tu.label, 'F1' : areas['AC1'], 'F2' : areas['AC2'], 'F3' : areas['AC3'], 'F4'
    print ('-----------------------------------------------------')

normalized = {}

normalized['FC1'] =np.float64(af['F1']-af['F1'].values.min() )/(af['F1'].values.max()-af['F1'].values.m
```

```
normalized['FC2'] =np.float64(af['F2']-af['F2'].values.min() )/(af['F2'].values.max()-af['F2'].v
normalized['FC3'] =np.float64(af['F3']-af['F3'].values.min() )/(af['F3'].values.max()-af['F3'].v
normalized['FC4'] =np.float64(af['F4']-af['F4'].values.min() )/(af['F4'].values.max()-af['F4'].v
normalized['FC5'] =np.float64(af['F5']-af['F5'].values.min() )/(af['F5'].values.max()-af['F5'].v
normalized['FC6'] =np.float64(af['F6']-af['F6'].values.min() )/(af['F6'].values.max()-af['F6'].v
normalized['FH1'] =np.float64(af['F7']-af['F7'].values.min() )/(af['F7'].values.max()-af['F7'].v
normalized['FH2'] =np.float64(af['F8']-af['F8'].values.min() )/(af['F8'].values.max()-af['F8'].v
normalized['FH3'] =np.float64(af['F9']-af['F9'].values.min() )/(af['F9'].values.max()-af['F9'].v
normalized['FH4'] =np.float64(af['F10']-af['F10'].values.min() )/(af['F10'].values.max()-af['F10
normalized['FH5'] =np.float64(af['F11']-af['F11'].values.min() )/(af['F11'].values.max()-af['F11
normalized['FH6'] =np.float64(af['F12']-af['F12'].values.min() )/(af['F12'].values.max()-af['F12

df1 = DataFrame(normalized)
df1=df1.fillna(0)
df1

kmeans = KMeans(n_clusters=6).fit(df1.iloc[:,1:])
centroids = kmeans.cluster_centers_
labels = kmeans.labels_
sample_silhouette_values = abs(silhouette_samples(df1.iloc[:,1:],labels))
sample_silhouette_values

clas={}
clas['Cluster']=labels
clas['Distance']=sample_silhouette_values
clas['TUs']=af['Label']
df2 = DataFrame(clas)

df2

clust=0

for clust in range(6):
    print ('\n******************** PLEASE WAIT ARE TRYING TO FIND OUT FAULTY TUs **************
    x = df2[df2['Cluster']==clust]
    mean = np.mean(x['Distance'])
    std = np.std(x['Distance'])
    Th = mean + std
    Th = 0.75
    print ("Cluster ", clust, "Threshold ", Th)
    print x[x['Distance'] > Th ]
    print ('\n*********************************************************************************
```

References

[1] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[2] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[3] Dionysia Kolokotsa. The role of smart grids in the building sector. *Energy and Buildings*, 116:703–708, 2016.

[4] C Talon and R Martin. Next generation building energy management system - new opportunities and experiences enabled by intelligent equipment. Technical report, Navigant Research, 2Q 2015.

[5] Yang Zhao, Tingting Li, Xuejun Zhang, and Chaobo Zhang. Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future. *Renewable and Sustainable Energy Reviews*, 109:85–101, 2019.

[6] Tadj Oreszczyn and Robert Lowe. Challenges for energy and buildings research: objectives, methods and funding mechanisms. *Building Research & Information*, 38(1):107–122, 2010.

[7] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N Kavuri, and Kewen Yin. A review of process fault detection and diagnosis: Part

iii: Process history based methods. *Computers & chemical engineering*, 27(3):327–346, 2003.

[8] https://www.iso.org/iso-50001-energy-management.html, 2018.

[9] Antonio Martínez-Molina, Isabel Tort-Ausina, Soolyeon Cho, and José-Luis Vivancos. Energy efficiency and thermal comfort in historic buildings: A review. *Renewable and Sustainable Energy Reviews*, 61:70–85, 2016.

[10] Woohyun Kim and Srinivas Katipamula. A review of fault detection and diagnostics methods for building systems. *Science and Technology for the Built Environment*, 24(1):3–21, 2018.

[11] Reece Kiriu and Steven T Taylor. Automatic fault detection and diagnostics and hierarchical fault suppression in ashrae rp-1455. *ASHRAE Transactions*, 123, 2017.

[12] VSKV Harish and Arun Kumar. A review on modeling and simulation of building energy systems. *Renewable and Sustainable Energy Reviews*, 56:1272–1292, 2016.

[13] Francis Moran, Tom Blight, Sukumar Natarajan, and Andy Shea. The use of passive house planning package to reduce energy use and co2 emissions in historic dwellings. *Energy and buildings*, 75:216–227, 2014.

[14] Yuebin Yu, Denchai Woradechjumroen, and Daihong Yu. A review of fault detection and diagnosis methodologies on air-handling units. *Energy and Buildings*, 82:550–562, 2014.

[15] Zhiwei Gao, Carlo Cecati, and Steven X Ding. A survey of fault diagnosis and fault-tolerant techniques part i: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics*, 62(6):3757–3767, 2015.

[16] Saqaff A Alkaff, SC Sim, and MN Ervina Efzan. A review of underground building towards thermal energy efficiency and sustainable development. *Renewable and Sustainable Energy Reviews*, 60:692–713, 2016.

[17] SS Chandel, Aniket Sharma, and Bhanu M Marwaha. Review of energy efficiency initiatives and regulations for residential buildings in india. *Renewable and Sustainable Energy Reviews*, 54:1443–1458, 2016.

[18] PB Usoro, IC Schick, and S Negahdaripour. An innovation-based methodology for hvac system fault detection. *Journal of dynamic systems, measurement, and control*, 107(4):284–289, 1985.

[19] D Anderson, L Graves, W Reinert, JF Kreider, J Dow, and H Wubbena. A quasi-real-time expert system for commercial building hvac diagnostics. *ASHRAE Transactions (American Society of Heating, Refrigerating and Air-Conditioning Engineers);(USA)*, 95(CONF-890609), 1989.

[20] Srinivas Katipamula and Michael R Brambley. Methods for fault detection, diagnostics, and prognostics for building systems—a review, part i. *Hvac&R Research*, 11(1):3–25, 2005.

[21] Frédéric Magoulès, Hai-xiang Zhao, and David Elizondo. Development of an rdp neural network for building energy consumption fault detection and diagnosis. *Energy and Buildings*, 62:133–138, 2013.

[22] Saad Abdul Aleem, Nauman Shahid, and Ijaz Haider Naqvi. Methodologies in power systems fault detection and diagnosis. *Energy Systems*, 6(1):85–108, Mar 2015.

[23] Michael C Keir and Andrew G Alleyne. Dynamic modeling, control, and fault detection in vapor compression systems. Technical report, Air Conditioning and Refrigeration Center. College of Engineering . . . , 2006.

[24] Zheng O'Neill, Xiufeng Pang, Madhusudana Shashanka, Philip Haves, and Trevor Bailey. Model-based real-time whole building energy performance monitoring and diagnostics. *Journal of Building Performance Simulation*, 7(2):83–99, 2014.

[25] James Weimer, Seyed Alireza Ahmadi, José Araujo, Francesca Madia Mele, Dario Papale, Iman Shames, Henrik Sandberg, and Karl Henrik Johansson. Active actuator fault detection and diagnostics in hvac systems. In *Proceedings of the fourth ACM workshop on embedded sensing systems for energy-efficiency in buildings*, pages 107–114. ACM, 2012.

[26] Balaje T Thumati, Miles A Feinstein, James W Fonda, Alfred Turnbull, Fay J Weaver, Mark E Calkins, and Sarangapani Jagannathan. An online model-based fault diagnosis scheme for hvac systems. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 70–75. IEEE, 2011.

[27] TI Salsbury. A temperature controller for VAV air-handling units based on simplified physical models. *HVAC&R Research*, 4(3):265–279, 1998.

[28] TI Salsbury and RC Diamond. Fault detection in hvac systems using model-based feedforward control. *Energy and Buildings*, 33(4):403–415, 2001.

[29] SR Shaw, LK Norford, D Luo, and SB Leeb. Detection and diagnosis of hvac faults via electrical load monitoring. *HVAC&R Research*, 8(1):13–40, 2002.

[30] Leslie K Norford, Jonathan A Wright, Richard A Buswell, Dong Luo, Curtis J Klaassen, and Andy Suby. Demonstration of fault detection and diagnosis methods for air-handling units. *HVAC&R Research*, 8(1):41–71, 2002.

[31] Siyu Wu and Jian-Qiao Sun. A physics-based linear parametric model of room temperature in office buildings. *Building and Environment*, 50:1–9, 2012.

[32] S. Wu and J. Sun. Multi stage regression linear parametric models of room temperature in office buildings. *Building and Environment*, 56:69–77, 2012.

[33] Xue-Bin Yang, Xin-Qiao Jin, Zhi-Min Du, Yong-Hua Zhu, and Yi-Bo Guo. A hybrid model-based fault detection strategy for air handling unit sensors. *Energy and buildings*, 57:132–143, 2013.

[34] Daniel Schwingshackl, Jakob Rehrl, Martin Horn, Julian Belz, and Oliver Nelles. Model extension for model based mimo control in hvac systems. *Journal of Building Engineering*, 11:224–229, 2017.

[35] Jakob Rehrl, Daniel Schwingshackl, and Martin Horn. A modeling approach for hvac systems based on the lolimot algorithm. *IFAC Proceedings Volumes*, 47(3):10862–10868, 2014.

[36] Leo H Chiang, Richard D Braatz, and Evan L Russell. *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2001.

[37] Jeffrey Schein, Steven T Bushby, Natascha S Castro, and John M House. A rule-based fault detection method for air handling units. *Energy and buildings*, 38(12):1485–1492, 2006.

[38] J. Schein, S. T. Bushby, N. S. Castro, and J. M. House. Results from field testing of air handling unit and variable air volume box fault detection tools. 2003.

[39] Arthur Dexter, Jouko Pakanen, et al. *Demonstrating automated fault detection and diagnosis methods in real buildings*. Technical Research Centre of Finland (VTT), 2001.

[40] Jeffrey Schein and Steven T Bushby. A hierarchical rule-based fault detection and diagnostic method for hvac systems. *HVAC&R Research*, 12(1):111–125, 2006.

[41] A Freddi, G Ippoliti, M Marcantonio, D Marchei, A Monteriu, and M Pirro. A fault diagnosis and prognosis led lighting system for increasing reliability in energy efficient buildings. 2013.

[42] Fadi M Alsaleem, Robert Abiprojo, Jeff Arensmeier, and Gregg Hemmelgarn. Hvac system cloud based diagnostics model. 2014.

[43] Haitao Wang, Youming Chen, Cary WH Chan, and Jianying Qin. A robust fault detection and diagnosis strategy for pressure-independent vav terminals of real office buildings. *Energy and Buildings*, 43(7):1774–1783, 2011.

[44] Manuel Peña, Félix Biscarri, Juan Ignacio Guerrero, Iñigo Monedero, and Carlos León. Rule-based system to detect energy efficiency anomalies in smart buildings, a data mining approach. *Expert Systems with Applications*, 56:242–255, 2016.

[45] Sung-Hwan Cho, Hoon-Cheol Yang, M Zaheer-Uddin, and Byung-Cheon Ahn. Transient pattern analysis for fault detection and diagnosis of hvac systems. *Energy Conversion and Management*, 46(18):3103–3116, 2005.

[46] Michael R Brambley, Nicholas Fernandez, Weimin Wang, Katherine A Cort, Heejin Cho, Hung Ngo, and James K Goddard. Final project report: Self-correcting controls for vav system faults filter/fan/coil and vav box sections. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2011.

[47] Haitao Wang, Youming Chen, Cary WH Chan, and Jianying Qin. An online fault diagnosis tool of vav terminals for building management and control systems. *Automation in Construction*, 22:203–211, 2012.

[48] Junichi Shiozaki and Fusachika Miyasaka. A fault diagnosis tool for hvac systems using qualitative reasoning algorithms. In *Proceedings of the Building Simulation*, volume 99, 1999.

[49] Alfonso Capozzoli, Fiorella Lauro, and Imran Khan. Fault detection analysis using data mining techniques for a cluster of smart office buildings. *Expert Systems with Applications*, 42(9):4324–4338, 2015.

[50] Zhun Yu, Benjamin CM Fung, and Fariborz Haghighat. Extracting knowledge from building-related data—a data mining framework. In *Building Simulation*, volume 6, pages 207–222. Springer, 2013.

[51] Cheng Fan, Fu Xiao, and Chengchu Yan. A framework for knowledge discovery in massive building automation data and its application in building diagnostics. *Automation in Construction*, 50:81–90, 2015.

[52] Samuel R West, Ying Guo, X Rosalind Wang, and Joshua Wall. Automated fault detection and diagnosis of hvac subsystems using statistical machine learning. In *12th International Conference of the International Building Performance Simulation Association*, 2011.

[53] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[54] Chen-Fu Chien, Shi-Lin Chen, and Yih-Shin Lin. Using bayesian network for fault location on distribution feeder. *IEEE Transactions on Power Delivery*, 17(3):785–793, 2002.

[55] Won-Yong Lee, John M House, Cheol Park, and George E Kelly. Fault diagnosis of an air-handling unit using artificial neural networks. *Transactions American Society of Heating Refrigerating and Air Conditioning Engineers*, 102:540–549, 1996.

[56] Jian Liang and Ruxu Du. Model-based fault detection and diagnosis of hvac systems using support vector machine method. *International Journal of refrigeration*, 30(6):1104–1114, 2007.

[57] Arthur L Dexter and D Ngo. Fault diagnosis in air-conditioning systems: a multi-step fuzzy model-based approach. *HVAC&R Research*, 7(1):83–102, 2001.

[58] Zhun Jerry Yu, Fariborz Haghighat, and Benjamin CM Fung. Advances and challenges in building engineering and data mining applications for energy-efficient communities. *Sustainable Cities and Society*, 25:33–38, 2016.

[59] Cheng Fan, Fu Xiao, Chengchu Yan, Chengliang Liu, Zhengdao Li, and Jiayuan Wang. A novel methodology to explain and evaluate data-driven building energy performance models based on interpretable machine learning. *Applied Energy*, 235:1551–1560, 2019.

[60] Dan Li, Guoqiang Hu, and Costas J Spanos. A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis. *Energy and Buildings*, 128:519–529, 2016.

[61] Rolf Isermann. *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer Science & Business Media, 2011.

[62] Yang Zhao, Shengwei Wang, and Fu Xiao. Pattern recognition-based chillers fault detection method using support vector data description (SVDD). *Applied Energy*, 112:1041–1048, 2013.

[63] A Beghi, R Brignoli, L Cecchinato, G Menegazzo, M Rampazzo, and F Simmini. Data-driven fault detection and diagnosis for hvac water chillers. *Control Engineering Practice*, 53:79–91, 2016.

[64] Chao Shang and Fengqi You. A data-driven robust optimization approach to scenario-based stochastic model predictive control. *Journal of Process Control*, 75:24–39, 2019.

[65] Andrew J Sonta, Perry E Simmons, and Rishee K Jain. Understanding building occupant activities at scale: An integrated knowledge-based and data-driven approach. *Advanced Engineering Informatics*, 37:1–13, 2018.

[66] SA Vaghefi, MA Jafari, J Zhu, J Brouwer, and Y Lu. A hybrid physics-based and data driven approach to optimal control of building cooling/heating systems. *IEEE Transactions on Automation Science and Engineering*, 13(2):600–610, 2016.

[67] Gerhard Zucker, Andreas Sporr, Stefan Kollmann, Alexander Wendt, Lydia Siafara Chaido, and Andreas Fernbach. A cognitive system architecture for building energy management. *IEEE Transactions on Industrial Informatics*, 14(6):2521–2529, 2018.

[68] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *HotCloud*, 10(10-10):95, 2010.

[69] Reynold S Xin, Josh Rosen, Matei Zaharia, Michael J Franklin, Scott Shenker, and Ion Stoica. Shark: Sql and rich analytics at scale. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of data*, pages 13–24. ACM, 2013.

[70] Álvaro Brandón Hernández, María S Perez, Smrati Gupta, and Victor Muntés-Mulero. Using machine learning to optimize parallelism in big data applications. *Future Generation Computer Systems, 2017, Elsevier*, 2017.

[71] Gema Bello-Orgaz, Jason J Jung, and David Camacho. Social big data: Recent achievements and new challenges. *Information Fusion*, 28:45–59, 2016.

[72] A Widodo and B-S Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, 21(6):2560–2574, 2007.

[73] Maitreyee Dey, Manik Gupta, Soumya Prakash Rana, Mikdam Turkey, and Sandra Dudley. A pid inspired feature extraction method for hvac terminal units. In *Technologies for Sustainability (SusTech), 2017 IEEE Conference on*, pages 1–5. IEEE, 2017.

[74] Karl Johan Åström and Tore Hägglund. *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.

[75] M Emre Celebi and Kemal Aydin. *Unsupervised learning algorithms*, volume 9. Springer, 2016.

[76] Sergios Theodoridis and Konstantinos Koutroumbas. Pattern recognition. 2003. *Elsevier Inc*, 2009.

[77] Boris Mirkin. *Clustering for data mining: a data recovery approach*. Chapman and Hall/CRC, 2005.

[78] M Dey, M Gupta, M Turkey, and S Dudley. Unsupervised learning techniques for HVAC terminal unit behaviour analysis. In *The 2017 IEEE International Conference on Smart City Innovations (IEEE SCI 2017)*, San Francisco, USA, August 2017.

[79] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[80] Erich Schikuta. Grid-clustering: An efficient hierarchical clustering method for very large data sets. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference On*, volume 2, pages 101–105. IEEE, 1996.

[81] Yannis Papanikolaou, Grigorios Tsoumakas, and Ioannis Katakis. Hierarchical partitioning of the output space in multi-label data. *Data & Knowledge Engineering*, 2018.

[82] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[83] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[84] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[85] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[86] Achmad Widodo and Bo-Suk Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, 21(6):2560–2574, 2007.

[87] Maitreyee Dey, Soumya Prakash Rana, and Sandra Dudley. Smart building creation in large scale hvac environments through automated fault detection and diagnosis. *Future Generation Computer Systems*, 2018.

[88] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[89] S P Rana, M Dey, H Siddiqui, G Tiberi, M Ghavami, and S Dudley. UWB localization employing supervised learning method. In *IEEE International Conference on Ubiquitous Wireless Broadband ICUWB 2017*, Salamanca, Spain, September 2017.

[90] Soumya Prakash Rana, Maitreyee Dey, Robert Brown, Hafeez Ur Siddiqui, and Sandra Dudley. Remote vital sign recognition through machine learning augmented uwb. 2018.

[91] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[92] Zhe Zhou, Chenglin Wen, and Chunjie Yang. Fault isolation based on k-nearest neighbor rule for industrial processes. *IEEE Transactions on Industrial Electronics*, 63(4):2578–2586, 2016.

[93] Li-Yu Hu, Min-Wei Huang, Shih-Wen Ke, and Chih-Fong Tsai. The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, 5(1):1304, 2016.

[94] Ming Ouyang. Knn in the jaccard space. In *High Performance Extreme Computing Conference (HPEC), 2016 IEEE*, pages 1–7. IEEE, 2016.

[95] Gautam Bhattacharya, Koushik Ghosh, and Ananda S Chowdhury. An affinity-based new local distance function and similarity measure for knn algorithm. *Pattern Recognition Letters*, 33(3):356–363, 2012.

[96] Hariharan Naganathan, Wai Oswald Chong, and Xuewen Chen. Building energy modeling (bem) using clustering algorithms and semi-supervised machine learning approaches. *Automation in Construction*, 72:187–194, 2016.

[97] Maitreyee Dey, Soumya Prakash Rana, and Sandra Dudley. Semi-supervised learning techniques for automated fault detection and diagnosis of hvac systems. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 872–877. IEEE, 2018.

[98] Soumya Prakash Rana, Maitreyee Dey, Mohammad Ghavami, and Sandra Dudley. Signature inspired home environments monitoring system using ir-uwb technology. *Sensors*, 19(2):385, 2019.

[99] Paul G Hoel et al. Introduction to mathematical statistics. *Introduction to mathematical statistics.*, (2nd Ed), 1954.

[100] Chidanand Apté, Fred Damerau, and Sholom M Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems (TOIS)*, 12(3):233–251, 1994.

[101] Chang Huai You, Kong Aik Lee, and Haizhou Li. An svm kernel with gmm-supervector based on the bhattacharyya distance for speaker recognition. *IEEE Signal processing letters*, 16(1):49–52, 2009.

[102] Soumya Rana, Javier Prieto, Maitreyee Dey, Sandra Dudley, and Juan Corchado. A self regulating and crowdsourced indoor positioning system through wi-fi fingerprinting for multi storey building. *Sensors*, 18(11):3766, 2018.