Automated Terminal Unit Performance Analysis Employing x-RBF Neural Network and Associated Energy Optimisation - A Case Study Based Approach

Maitreyee Dey*[a], Soumya Prakash Rana[a], Sandra Dudley[a]

[a]*School of Engineering,*
London South Bank University, London, United Kingdom

Abstract

An artificial neural network based framework, to analyse and address key energy related performance issues inside building environment is proposed. The study engages thousands of heating, ventilation, and air conditioning terminal unit data from real commercial buildings employing a new feature engineering method augmenting terminal unit time-series data to create novel features. These features are subsequently fed into the proposed neural network, named x-RBF which is a combination of x-means clustering followed by radial basic function neural networks for automatic fault detection and diagnosis. The new model has been successfully employed and investigated on different types of heating-cooling patterns concerning power demand and control strategies from actual building historical terminal unit data. The proposed x-RBF model has been trained-tested on approximately three years of building data and achieves 99.7% sensitivity, a first for real building applications. Comparison has been made with other neural networks to verify the performance of the proposed x-RBF and it is further validated through statistical measurements. The proposed model demonstrates its ability to truly predict and anticipate fault scenarios in terminal unit operation. Consequently, energy and cost calculations have been executed to realise the potential financial impact (as a consequence of performance improvements) that can be achieved by the proposed framework in the building environment.

Keywords: Heating, Ventilation and Air-Conditioning, Terminal Unit, Artificial Neural Networks, Automatic Fault Detection and Diagnosis, Energy Optimisation

1. Introduction

Remote observation and fault identification can provide impactful augmentation to existing building energy management systems (BEMS), also known as building automation systems (BAS). Implementing such extensions create scalable and future-proof schemes for improved building management, building understanding, energy use optimisation, and post-occupancy energy-gap reductions. This approach effectively leads to the empowerment of building designers, owners, and post-occupancy building

managers, who do not always have experience of the nuances of the resident Heating, Ventilation and Air-Conditioning (HVAC) systems within the buildings they manage or reside. The HVAC systems are those elements closest to the building user and as such have the greatest impact on user comfort and well-being, building problem registrations and complaints. Remote problem identification and resolution tracking not only leads to improved building management and sector impact, but also provides further benefits such as occupier well-being improvements, operational cost savings, greater appreciation and understanding of human-building interaction (HBI), suitable renewable energy retro-fit benefits and choices, building to building interactions for future smart cities and building design developments. Automatic identification of faulty unit behaviours (affects heating and cooling operation) from real BEMS data is notoriously difficult with any action generally involving expert building engineers and their knowledge to interpret the detected faults of a unit within the building HVAC system. Manual data investigation and fault-finding processes are haphazard, time consuming, require ad-hoc expert knowledge and experience, which is slow and expensive and can lead to eventually losing control of the many problems within buildings. Hence, bringing automation and intelligence to this process, using data mining and machine learning (ML) is an ideal solution. The large volume of data is the most powerful component and foundation towards any appropriate, general, decision-making system. Thus, automation in the fault detection and diagnosis (FDD) study relies heavily on real environments with real sensor data, which can be used to analyse the performance of corresponding electrical units, defined as automatic fault detection and diagnosis (AFDD). Hitherto, there is no standard established for efficient energy management in buildings, even though building sectors are responsible for 40% of global energy consumption with heating, ventilation, and air-conditioning (HVAC) systems of residential and commercial buildings alone, responsible for approximately 39% to 41% of the total building energy consumption [1]. Therefore, HVAC is a prime focus of building researchers to improve user comfort and energy efficiency. The HVAC is a complex system comprising several units such as, air handling unit (AHU), variable air volume (VAV), chiller, boiler, fan coil unit (FCU). Usually, commercial buildings have one or two major units i.e., chiller or boiler but hundreds to thousands of fan coil units are fitted inside depending on the size of the building. Thus, if a single small unit is not working desirably, it can affect the entire plant. This study has been narrowed down to consider fan-coil unit's terminal unit (TU) (displayed in Fig. 1), to monitor their performance and auto-identify malfunctioning units. Fig. 1 shows the overlapping and growing relationship of TU with surrounding layers within BEMS, and how these services interrelate each other.

2. Background

Fault detection is the study of recognising problems that have occurred, even if their cause is unknown. Fault diagnosis indicates the causes of these problems, to discover points where corrective action can be taken. The AFDD processes have made

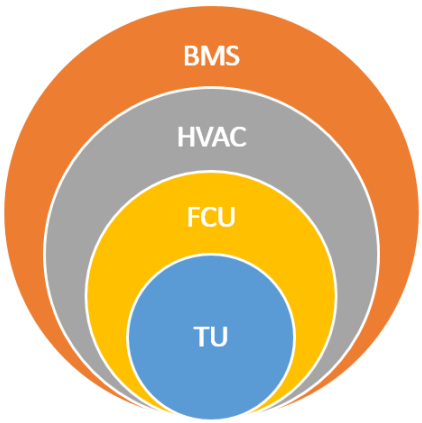


Figure 1: An overlapping relationship diagram of the building's terminal unit within BEMS.

notable advancements combining data mining and ML techniques. Dynamic research and exploration in this field began in the 1980s [2], however, from the outset practical limitations such as scalability and the complexity of HVAC systems have made FDD extremely challenging. The area of FDD can be categorised into several groups depending on their applications as reported in the literature. There are three categories; the quantitative model, the qualitative model, and history process-based approaches of FDD reported in approximately ninety articles prior to 2005 [3]. The FDD methods are further categorised into a further three groups in [4]: analytical methods, knowledge-based methods, and data-driven methods. One recent article by Kim et. al reviews two hundred articles published on classification of FDD methods [5]. The article states that 62% of published research works are based on historical process, 26% research works cover qualitative models and 12% are performed on quantitative models. The studies performed on historical processes use measured data obtained over certain time periods to derive fundamental knowledge. The process history-based approaches are the most effective ways of analysing large-scale data to identify recurring patterns and hidden knowledge to automatically detect faults. It is realised from the recent literature that applied machine learning has huge potential to identify the faults automatically however, it requires historical data and appropriate methodologies to learn a building heating and cooling trends [1]. Gunay et. al. performed a critical review on AHU and VAV terminal units for commercial buildings carried out over the last 20 years [6]. In this paper common building faults, along with proposed strengths and weaknesses of the reported research is included. Building operation and control while dealing with FDD including the procurement of system also has been described in [7]. Recent advancements has enabled AFDD to be further classified into two sub-categories: (a) knowledge based and (b) data-driven based. It is worth mentioning from the previous research that knowledge-based analysis requires vast knowledge from domain experts [1], which can be difficult to accumulate. Therefore, the proposed work has been performed to characterise TU behaviour employing data-driven approach to

avoid the limitations of the knowledge-based approach, provisioning a more general outcome. Further, other related data driven AFDD approaches have been described in the following section.

2.1. Data-driven based FDD methods

Data-driven relates to the approaches which derive essential information from the historical data (or stored data, or available data) employing data mining algorithms to identify and diagnose abnormalities [8, 9]. Thus, data-driven approaches for BEMS have become increasingly important in the era of big-data research to capture non-linear, complicated relationships, and forecast system behaviour using ML techniques. However, it is hugely challenging for building professionals to infer behavioural information, understand their meaning and to trust predictions, as the developed models are highly complex and have low interpretability for end users. Magoules et. al. proposed an artificial neural network (ANN) prototype implementing recursive deterministic perceptron (RDP) to facilitate a FDD for an entire building. This method aimed to detect faults and ranking equipment that were more defective or risky units first [8]. In [9], k-means clustering based data-driven model is proposed for three AHUs operation analysis. One of these methods is proposed by Fan et. al., to describe and assess a data-driven based building energy performance evaluation model to assist building engineers by interpreting upcoming HVAC behaviours [10]. A new metric, i.e., 'trust' is proposed by Li et. al., which is an alternative approach to the conventional accuracy metric to evaluate model performance via operational building data. Conversely, the method establishes a relationship between HVAC data patterns and faulty categories of a system [11]. The data-driven based methods extract the key components of data by transforming and projecting them into new dimensions. Subsequently, the key components are used instead of the whole dataset aiming to represent individual HVAC behavioural patterns in more meaningful way, helping to identify faults efficiently. This approach suits modern HVAC systems employed in commercial buildings, which mainly contains three phases, statistical understanding of building data, feature engineering, FDD employing ML algorithms. Building parameter's data (e.g. heating-cooling temperature, power) are usually time series data thus wavelet and short-time Fourier analysis are common feature engineering methods in these cases [12]. A combined approach of wavelet transformation (WT) and principal component analysis (PCA) is proposed by Zhao et al. to detect faults for the AHU system, where the time series data are transformed using WT followed by dimensionality reduction procedure via PCA to map the large data into lower dimensional space. Later, lower dimensional data are used to predict faults [13]. In practice, this FDD methodology is appropriate for fault detection rather than fault diagnosis, thus, hybrid procedures are considered for efficient FDD systems in large buildings [14]. Recent work published by Shag et. al. developed a stochastic model predictive control (SMPC) aimed to provide a promising solution for complex control problems under uncertain disturbances. In this paper, the SMPC approach is proposed by actively learning an uncertainty dataset utilising a ML method [15]. Sonata et. al. modelled a prototype to understand and

classify occupant behaviour in a building to optimise the energy usage. Energy usage was collected from plug load sensors installed in an open-office building in San Francisco, USA. Further the data was classified to understand occupant's activity patterns and forecast the building energy usage, taking necessary energy management steps [16]. A statistical approach using generalised Cochran-Orcutt estimation is employed by Vaghe et. al. to adjust a linear forecasting model and handle a building's total energy consumption. It 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 simulation software. However, this method is limited to a fixed term dataset which needs to adapt appropriately for new and longer periods of data [17].

The range and depth of the literature demonstrates that the BEMS research domain and in particular HVAC systems are extremely challenging and vast, typically monitoring a unit's performance in real-time [18] Most of the AFDD methods were developed to solve problems related to specific large units such as, chillers, boilers, AHU, and VAV [19, 20]. However, none of them focus on small units i.e., fan-coil terminal units although these small units have a huge impact on the building's energy consumption with building energy consumption prediction attracting much attention from the research community. Comparative analysis of the commonly used methods (such as, Elaborate eng., Simplified eng., SVMs, ANNs, statistical.) for the prediction of building energy consumption has been described in [21]. However, numerous open, unsolved research problems remain. Many energy related problems are left undetected and ignored due to the diverse and complex energy systems installed and uncertainty of occupant behaviour. Thus, controlling and monitoring each unit and detecting when the faults occurs is burdensome for building managers (whether building experts or not) or the building service provider. Additionally, much of the research has been performed on model-generated data, and not on real building data. This along with a global lack of agreed standardised methods available to deal with building management plans related to faults detection and subsequent actions regulations. Thus, the objectives of this research are: 1) the proposal and development of a robust method for automatic and fast identification irrespective of the fault type, 2) testing on large scale, real buildings to overcome human limits on processing the vast data involved 3) endorsement of the proposed method by expert building engineer and via statistical methods, and 4) optimise energy consumption within the case study building.

2.2. Real challenges of the terminal unit

Airflow and 'comfortable' temperature requirements depend on many factors that may be internal or external to a real building. The internal factors might be device related issues (i.e., damper stuck at full close position or partial position, zone temperature sensor reading frozen, air flow sensors reading frozen, low supply air static pressure, etc.) all causing inefficient temperature control. Other external causes such as those caused by occupants or geographical positioning (i.e., open window, manual changing of temperature set points, room position in shadow/ sunny, occupant inhabitancy, etc.)

also affect control strategies in similar ways. 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 - 21^o C and whenever the room temperature goes beyond these limits, the respective heating or cooling power is executed) resulting in action termed 'hunting' TU behaviour. The cooling and heating set points are generally setup by the building operators initially, depending on seasons and the weather, in winter it can be set between 23^o - 27^o C and in summer 18^o - 22^o C. In an office space, if the TU is on beyond working hours this is type of scheduling error resulting in unnecessary electricity usage with additional power demands subsequently evident to maintain the room temperature at night (which was only meant for daytime use). Room and sensor location are also a factor for abnormal behaviour (e.g., 17^o C ambient temperature requests in a room that has continuous sunlight exposure or is "settled" at 26^o C ambient temperature on a summer day or where the sensor is fixed near to refrigerator room resulting in false ambient temperatures). These faults must be handled by the expert building engineers daily through the service provider platform. 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 classify each one of these issues as they occur or identify malfunctions observed to summate individual faults. It has been found during the proposed case study that only 50% (approximately) of faults have been resolved among all logged/reported issues, causing vast energy wastage, problem backlogs and demonstrating how issues soon become unmanageable. It indicates that so many problems are identified late and left unresolved due to a lack of automation and intelligence in BMS. Thus, irrespective of any specific type of fault, any type of misbehaving TU must be addressed and categorised at this stage of the research. In addition, the same TU may behave differently on different days, which is considered as separate behaviour and must be addressed when making a general solution. It is difficult to discover the causes of every single faulty TU by direct observation (for building engineers) to make the ground truth. Fig. 2 shows two real TU behaviours from the case study building comprising one faulty and one non-faulty for a single day (24hrs), where the blue line is the non-faulty TU, and the red line is the faulty and depicts hunting behaviour.

2.3. Contribution

The proposed work creates a new two-phase intelligent solution to address faulty HVAC TU operation in smart buildings, employing machine learning. The aim of this work is to bridge the gap between on-going research, its implementation on real building's energy conservation and to optimise its usage. This is a data-driven approach, able to process and analyse behaviour in such a way that can help the building engineer to diagnose the problem, the TU and fix it sooner than before. This work is modified to overcome the limitation of the previously presented work by the authors in [22, 23]. The collected time-series energy data are non-linear in nature. thus, this work combines X-means clustering followed by Radial basis function (RBF) neural network resulting in the proposed x-RBF neural network method. This x-RBF method demonstrates

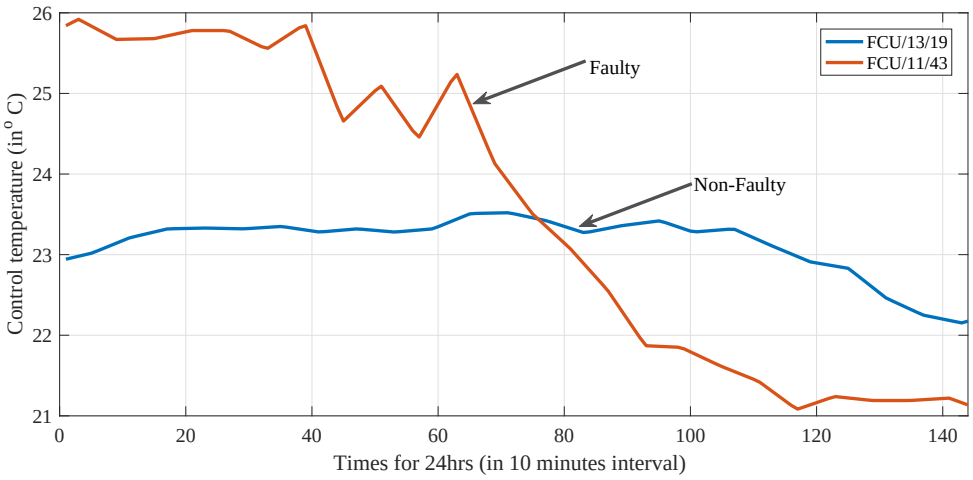


Figure 2: Examples of faulty and non-faulty control temperature.

excellent robustness and improved fault detection accuracy over the previously proposed method. The x-RBF have been developed to perform both the clustering and classification tasks, respectively. The main contributions of the proposed work are as follows:

(a) A new feature engineering method has been developed with pseudo labelling also being performed through verification by expert building engineers to realise the energy data patterns and for auto-recognition of TU faults.
(b) A new collaborative framework employing X-means and RBF is proposed as x-RBF, to implement the proposed model with less computational complexity.
(c) Two other neural networks; Probabilistic neural network (PNN) and Generalized regression neural network (GRNN) have been chosen based on their non-linear classification ability for comparison with the proposed method.
(d) The proposed algorithm has been implemented on two real large-scale commercial building's data to understand its effectiveness in accurately recognising the unusual events that effect thermal comfort and unwanted power demands.
(e) The performance of this new learning inspired automatic TU fault identification method has been analysed through statistical metrics and the outcomes described by case studies.
(f) The energy consumption by the TUs and its associated cost have been estimated to show the impact of the proposed method towards energy conservation and its optimal usage if applied in the building energy sector.

3. Proposed Methodology

The AFDD framework of the entire proposed system is comprises of two parts: the building data accumulation and the proposed AFDD which is shown in Fig. 3.

The figure presents the stepwise development of each module such as, data storage, retrieval, analysis, visualisation, and decision-making. The details are described in the subsections below.

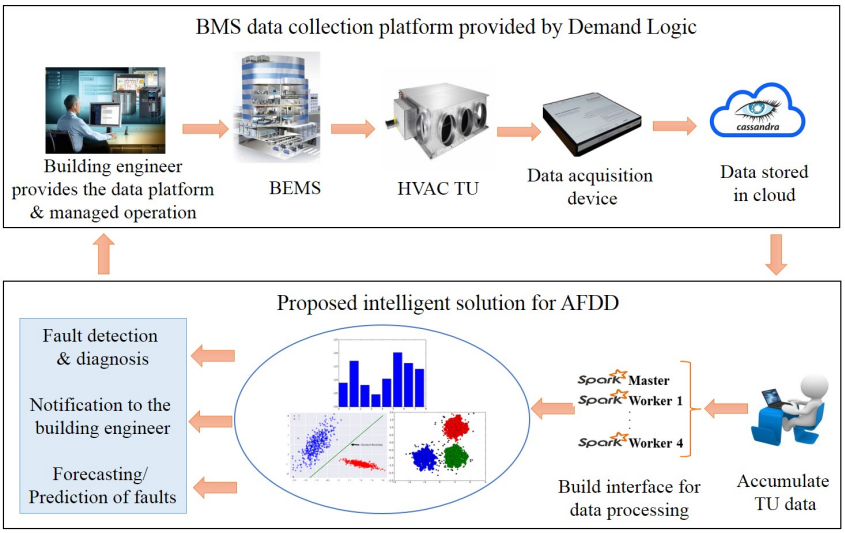


Figure 3: The framework of the system architecture for this experiment.

3.1. Building data accumulation

Initially, the building data is transferred from the source to a cloud internet through a secured gateway on the dedicated BMS network. A data acquisition device (DAD) has been used to collect the data (installed by project partner company Demand Logic Ltd.) [22], shown in Fig. 3. The DAD collects several unit's data, but this work is focused upon TU's performance analysis thus only TU data has been considered for further processing. This framework provides access to pivotal information along with precise time instances making it suitable for pre-processing. Once the data pre-processing (including missing data interpolation) has been performed, the data patterns and their associated parameters behaviour must be analysed.

3.1.1. Overview of TU data acquisition and understanding

The HVAC system provides three key facilities: heating, ventilation, and air-conditioning to deliver thermal comfort and provide satisfactory indoor building air quality. The TU is a sub-unit of the HVAC system, it's a simple device comprising a heating and/or cooling heat exchanger and fan. It is generally wall or ceiling-mounted and monitored via thermostats inside buildings. There is a central chiller and boiler plant connected in the building that distributes cold water to all the cooling coils and hot water to all the heating coils. The thermostat senses and signals the corresponding water valve depending upon the environmental requirements, cold water is passed through the coil and recirculates the hot air by the fan if the room gets too warm or

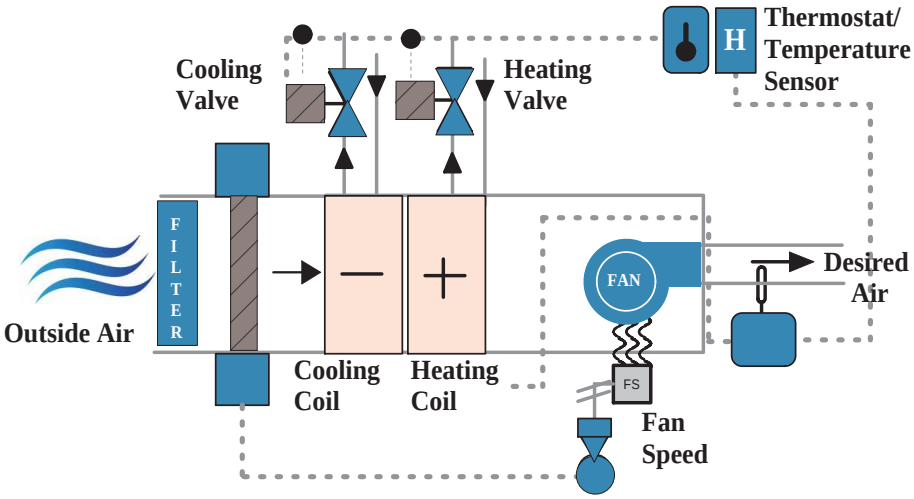


Figure 4: A schematic diagram of fan coil unit.

distributes warm air if the room becomes too cold. A schematic of TU device is shown in Fig. 5.

The six TU parameters are specifically selected depending upon their significance to generate the feature vector given in the Table 1. All the parameters were gathered in 10 minutes intervals.

Table 1: The list and description of the TU parameters.

Parameters	Description	Units
Control temperature	Room temperature/ Return air temperature	Degree centigrade (^{o}C)
Cooling set-point	Temperature that maintains cooling in the space of the zone/ room	Degree centigrade (^{o}C)
Heating set-point	Temperature that maintains heating in the space of the zone/ room	Degree centigrade (^{o}C)
Cooling power	The measured capacity of the cooling system to remove the heat.	Kilowatt (kW)
Heating power	The measured capacity of the heating system to heat up the space.	Kilowatt (kW)
Enabling signal	This allows passing through input signal and measures the system's status.	N/A

3.1.2. Potential issues in TU operation

Several potential faults such as hunting, on-ness, and saturation have been studied for various TUs to identify the most prevalent issues [22, 23, 24]. (i) Hunting is measured by how much the control temperature fluctuates over a day from its set point. (ii) On-ness is considered the time when a TU has continuous cooling or heating power demands throughout the day including its operational hours. (iii) Saturation is

the proportion of time over a day that the valve or damper is open at its maximum. Thus, the higher the value, the longer heating or cooling valve (or damper) is open, resulting in continuous power demand. These potential faulty TU behaviours occur for several physical reasons which are listed below:

(a) Stuck-valve — this is indicated by a saturation value of 1, which means the valve has been fully open over whole day.
(b) Poor sensor location — location of the temperature sensor at an non-optimal position i.e., back of a drinks cabinet (in close proximity to heating and/or cooling elements).
(c) Out of operational hour- a unit may function beyond operational hours due to either it has been forgotten (manually operated and left) or the input of the incorrect operational time schedule.
(d) Imperfect control — this is generally due to the settings of tight deadbands or over dynamic PID control. This may result in frequently switching between heating and cooling demands.
(e) Unachievable set point — often it happens that occupants manually change the set-point to a value that is unachievable for that environment.
(f) Unable to receive sufficient flow or upstream temperatures — the flow temperature from the boiler or chiller is not strong enough for the space, often because of over-demanding temperature.
(g) Competition from nearby units- sometimes due to the physical location of two adjacent TUs they affect each other's operation. A TU may try to cool the space whereas another adjacent TU tries to warm the same area. Generally, it is found where there is poor hierarchical control across a group of TUs.
(h) Varying set point — typically, the set point may be changed rapidly because of occupant's thermal discomfort (e.g. occupant's change set point depending on their personal preference).
(i) Location effects — either due to high solar gains or position of TU is very close to energy-consumption equipment with high internal gains, e.g. an old lighting fixture or photocopier can cause unusual thermal demands.
(j) Incorrect sizing for true demand — sometimes, he load is underestimated, and a larger unit should have been fitted. This scenario is more often found in cooling mode than heating.

Although a TU is considered a "simple device", a multitude of issues can lead to faulty operation/ behaviour requiring professional building expert knowledge to recognise each one of these issues. Manual TU investigations are tedious and becoming impossible with ever-increasing building data and the limitation of qualified resident building engineers. Hence automating and bringing intelligence to this process would be an ideal solution. To deal with these aforementioned issues a novel, user friendly TU data-driven approach has been proposed to identify different TU behavioural patterns without the need of an expert building engineer towards making an automatic fault detection and diagnosis system for energy optimisation.

3.2. Proposed AFDD approaches

The proposed AFDD system for building's TU data primarily consist of four phases, summarised in the flow diagram below.

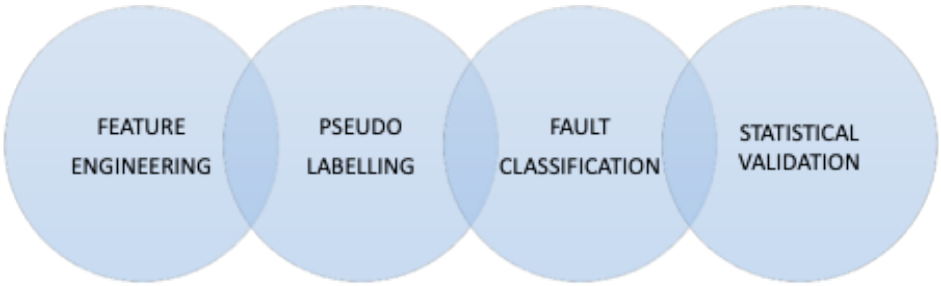


Figure 5: The flow diagram of the proposed AFDD approach.

3.2.1. Feature engineering

Building TU data are high-dimensional data streams, which are inherently difficult to use and are non-trivial for analysis purposes. To manage this problem an 'intelligent' feature engineering method is proposed to manage with the available high-dimensional TU data [22, 23]. It is an 'attribute reduction process', which projects datasets of higher dimensions into a smaller number of dimensions intended to be informative, non-redundant, facilitating subsequent learning and improved automation. The data-driven feature engineering method is accomplished by employing six TU parameters: control temperature, set point, dead band, heating effort, cooling effort, and enabling signal. These time series data are collected at 10-minute intervals from real buildings throughout the day by a secured BMS network gateway. The proposed feature engineering method has been executed considering three different events from the area under control temperature and corresponding power curves of a single TU data. The events are divided into three different stages: (1) event discovery, (2) event area calculation, and (3) event aggregation stage for each day (24hrs), and described as follows:

(1) **Event discovery stage:** Event discovery is inspired by the proportional integral derivative (PID) controller response curve [25]. A PID controller accurately handles the process variables to control processes, remove oscillations, and increase the industrial control system efficiency. Likewise, when the operational hour starts, the temperature begins to adjust depending upon the building's environmental requests. Hence, the control temperature variation with the set point is divided into four stages: event start, response delay, goal achieved, and event end, based on the response curve shown in Fig. 6. In Fig. 6, the pink coloured area indicates the heating event, the blue coloured area indicates a cooling event for both the control

temperature (in degree centigrade) and associated power demands (in kilowatt) of a whole day (for 24 hrs). The different events for the control temperature and power are indicated in Fig. 6.

(a) Event start (ES): An event start is assumed to happen when the BMS is first switched on in a day (when enabling signal gets switched on).

(b) Response delay (RD): The temperature starts to respond only after a certain delay from the previous point when the BEMS is switched on (due to the process variable delay during the dead time) and this is termed as response delay.

(c) Goal achieved (GA): A goal achieved event is assumed to happen when the control temperature reaches the desired set point. The GA is considered as the time instant when the process variable reaches the steady state, or final value.

(d) 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.

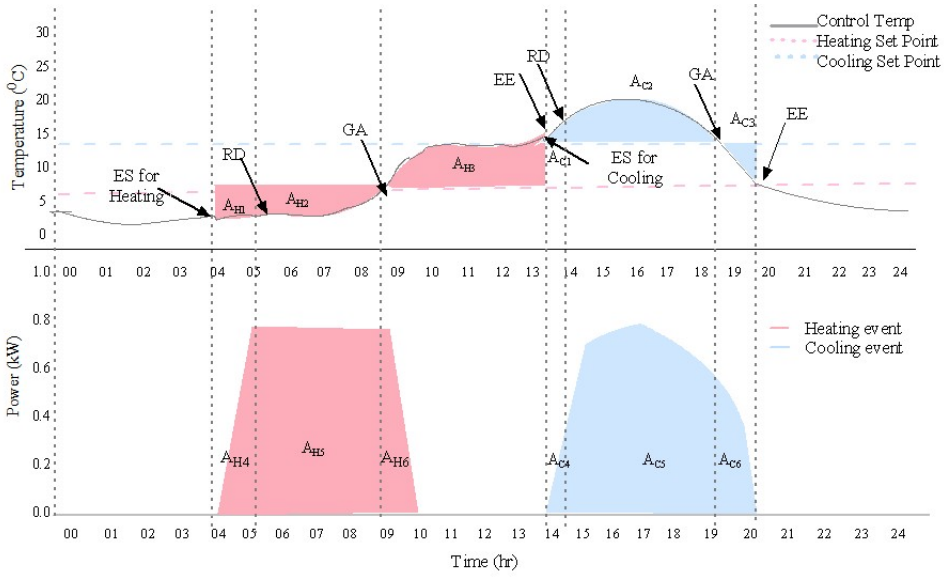


Figure 6: Event discovery process from a single day TU data.

(2) **Event area calculation stage:** Once the suitable heating and cooling events have been determined, the area below the temperature and power lines for every event is measured. Based on the type of events, area calculations are performed respectively for both the 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 measured. Total twelve distinct areas are derived from a single day TU data. Eq. (1) depicts the area (A_E) under the curve $f(x)$ at every time interval Δx.

$$A_E = \sum_{i=0}^{n} f(x_i)\Delta x \tag{1}$$

For an event of heating, the calculations for temperature areas are denoted by A_{H_1} *to* A_{H_3} and for power by A_{H_4} *to* A_{H_6}. Similarly, for a cooling event, the area calculations for temperature are denoted by A_{C_1} *to* A_{C_3} and for power are by A_{C_4} *to* A_{C_6}. Once these areas have been computed, it has been normalized to obtain the final feature values as F_{H_1} *to* F_{H_6} and F_{C_1} *to* F_{C_6} as indicated by Eqs. (2) to (5).
Eqs. (2) and (3) show the area calculations for a heating event.

$$\begin{aligned} F_{H_1} = \frac{A_{H_1}}{T_{H_1}}, \quad F_{H_2} = \frac{A_{H_2}}{T_{H_1}}, \quad F_{H_3} = \frac{A_{H_3}}{T_{H_2}} \\ where, \;\; T_{H_1} = max(A_{H_1} + A_{H_2}) \\ and, T_{H_2} = max(A_{H_3}) \end{aligned} \tag{2}$$

$$\begin{aligned} F_{H_4} = \frac{A_{H_4}}{P_{H_1}}, \quad F_{H_5} = \frac{A_{H_5}}{P_{H_1}}, \quad F_{H_6} = \frac{A_{H_6}}{P_{H_2}} \\ where, \;\; P_{H_1} = max(A_{H_4} + A_{H_5}) \\ and, P_{H_2} = max(A_{H_6}) \end{aligned} \tag{3}$$

Eqs. (4) and (5) show the area calculations for a cooling event.

$$\begin{aligned} F_{C_1} = \frac{A_{C_1}}{T_{C_1}}, \quad F_{C_2} = \frac{A_{C_2}}{T_{C_1}}, \quad F_{C_3} = \frac{A_{C_3}}{T_{C_2}} \\ where, \;\; T_{C_1} = max(A_{C_1} + A_{C_2}) \\ and, T_{C_2} = max(A_{C_3}) \end{aligned} \tag{4}$$

$$\begin{aligned} F_{C_4} = \frac{A_{C_4}}{P_{C_1}}, \quad F_{C_5} = \frac{A_{C_5}}{P_{C_1}}, \quad F_{C_6} = \frac{A_{C_6}}{P_{C_2}} \\ where, \;\; P_{C_1} = max(A_{C_4} + A_{C_5}) \\ and, P_{C_2} = max(A_{C_6}) \end{aligned} \tag{5}$$

(3) **Event aggregation stage:** Various heating and cooling events may occur during an entire day, thus all the events of a given heating and cooling type must be aggregated to represent the averaged values. A further step in the feature engineering process is event aggregation. The final combined features can be presented by Eqs.(6) and (7), where k signifies the event number and n denotes total number of occurrences for each event in both the heating and cooling event type. Thus, a

whole day TU data can be presented by twelve features for both the heating and cooling events. This stage generates the final feature vector for machine learning application to identify different fault from these TU data.

$$F_{H_k} = \frac{1}{n}\sum_{i=1}^{n}(F_{H_{k_i}}) \tag{6}$$

$$F_{C_k} = \frac{1}{n}\sum_{i=1}^{n}(F_{C_{k_i}}) \tag{7}$$

3.2.2. Pseudo labelling with X-means

Due to the absence of prior information about the TU data involved in this investigation, an unsupervised learning methodology is employed to discover a set of TU behavioural patterns. The clustering algorithm learns the underlying data pattern and groups the data without knowing any physical information about the TU data. However, the effectiveness of the clustering relies on the data distribution and presumption of cluster number [26, 27]. As, the types of TU behaviours (number of clusters) are unknown it is difficult to discover appropriate clusters from conventional clustering methods that represent the distinct behaviours of TUs. Thus, the investigation employed the extended K-Means (X-means) [28, 29] algorithm, an appropriate approach in this context comprising two steps: Bayesian information criterion (BIC) (defined in Eq. (9)) followed by the k-means algorithm (defined in Eq. (8)) with different cluster seed values. The BIC phase helps to identify the optimal number of clusters gathered for the TU dataset and then clusters them based on the proximity measure using k-means, defined below.

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n}||X_i^{(j)} - \mu_j||^2 \tag{8}$$

Where, $||X_i^{(j)} - \mu_j||^2$ is a Euclidean distance measure between a TU behaviour (in a day) $X_i^{(j)}$ and the cluster centre μ_j (represents a behaviour). Each TU is assigned to that group which is closest to the centroid. After all the TUs (data points) are assigned to the corresponding groups, the positions of the k centroids are recalculated. The above steps are reiterated till the centroids no longer move.

$$\begin{aligned} BIC_{score} &= -2\log(L) + K\log(n) \\ where, L &= P(x \mid \theta, M) \end{aligned} \tag{9}$$

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 TU data, the parameter of the model, total number of data points and the number of free parameters to be estimated, respectively. Once the centroid (k) is determined and k-means performed, K_{max} is then

selected, and all centroids tested. The BIC score is then compared for K and K_{max} as the lower BIC score is always preferred for better fitness to the data. If the current model obtains a better score, then the split is considered the best strategy for clustering and the cluster number is decided here, which is six in this case (for both case studies). Each of these clusters represent a distinct behaviour of TU characteristics (good/bad) and have been further applied for pseudo labelling of all the functioning TUs. A pseudo code for the pseudo labelling implementation has been displayed in the Algorithm 1.

Algorithm 1 Proposed pseudo labelling

Require: Power and temperature's TU features = $\{f_1, f_2, ..., f_n\}$

1: $Maximum\ number\ of\ cluster = Max$
2: $K = Previous\ model's\ BIC\ score$
3: $K_{max} = Present\ model's\ BIC\ score$
4: **for all** $max = 2$ *to* 10 **do**
5: $Initial\ mean\ values\ assigns = \mu_1, \mu_2, ..., \mu_{max}$
6: $Assign\ each\ TU\ to\ the\ cluster\ which\ has\ closest\ mean \Leftarrow using\ the\ Eq.\ (8)$
7: $Update\ means$
8: $Calculate\ BIC\ score\ for\ current\ model\ K_{max} \Leftarrow using\ the\ Eq.\ (9)$
9: **if** $K > K_{max}$ **then**
10: **return** *Go to line* 5
11: **else**
12: $Max\ is\ considered\ as\ final\ cluster\ number\ \&\ exit\ loop$
13: **end if**
14: **end for**
15: $Six\ centroid\ for\ each\ cluster = \{\mu_0,\ \mu_1,\ ...,\ \mu_5\}\ are\ produced\ by\ X-means$
16: $Six\ groups\ of\ labelled\ patterns = \{C_0,\ C_1,\ ...,\ C_5\}\ are\ produced\ by\ X-means$

Thus, X-Means clustering has partitioned the daily TU data into six different groups (six distinct behavioural patterns) to represent the behaviour of TUs in both case studies 1 and 2. These six mathematically obtained groups have also been verified by expert building engineers prior to performing subsequent experiments. The six types of TU behaviours obtained from the clustering results are presented in Table 2. It provides an insight into the TU characteristics without any prior knowledge of identifying faulty and non-faulty trends. Each cluster is labelled as C_0, C_1, C_2, C_3, C_4, and C_5. The labels have been employed further to classify the TUs for automatic fault detection and diagnosis purposes developing perceptron learning to predict faults, discussed in the following section.

3.2.3. Fault classification with suitable ANN algorithms

The ability of the ANN to comprehensively process information follows from their capacity to generalise and isolate hidden dependencies between input and output data.

Table 2: Description of discovered TU patterns

Cluster number	Description of pseudo labels
C_0	The control temperature is achieved set-point with low power effort, working in within operating hours.
C_1	The control temperature is achieved set-point with medium or average power effort.
C_2	The control temperature is achieved set-point with high-power effort.
C_3	The control temperature is hunting patterns along with medium to high power effort and working outside operational hours.
C_4	The control temperature is does not achieved set-point (out of deadband by up to 5^oC) with high power effort.
C_5	The control temperature is does not achieved set-point (out of deadband by up to 10^oC) with high power effort.

A significant advantage of neural networks is that they are capable of learning and generalising accumulated knowledge and can predict on unobserved data. Most of the predictive tasks are characterised by high levels of non-linearity and non-stationary, noisiness, irregular trends, jumps, abnormal emissions. ANN does not impose any restrictions on the input variables, so it is adaptable with problem specific solutions and worth exploring as trial-and-error methods. ANN requires less statistical training than other conventional ML algorithms (such as, support vector machine (SVM), k-nearest neighbour (KNN), etc.) to make a decision boundary [30, 31]. Algorithms such as, SVM and KNN have been employed by the authors previous work [22], where AFDD resulted in low recall (identified the proportion of actual faulty TUs that were correctly detected). Thus, here ANNs have been explored and experimented for their adaptable nature. The ANNs comprise several layers making them more complex, but provide greater learning and prediction capability over the others mentioned. Conventional ANN, such as multi-layer perceptron (MLP) has drawbacks over time series data prediction where the data volatility is very high. Generally, MLP displays problems in the local minima because of its training function. Also, the number of hidden layer and nodes needed are fixed through trial and error, where the incorrect estimation causes under-fitting or over-fitting. Thus, the RBF based neural network has been employed to avoid these limitations [32], providing a more suitable method of AFDD for HVAC TUs. The RBF based networks are relatively easy to design, train and have a strong tolerance to noise, which makes them suitable for online training. Three different versions of RBF have been implemented here: conventional RBF, PNN, and GRNN to propose an efficient NN based AFDD system. These networks do not need to be experimented with different numbers of hidden nodes because the number of hidden nodes is decided when calculating cluster centroids. These have been already measured by the X-means clustering in the first phase of the proposed work and that are utilised here for the RBF hidden node calculation. The experimental results are discussed in the following sections.

(1) **Radial basis function network:** The RBF neural network is a fast classification process and a type of feed forward neural network consisting three layers: an input layer, a hidden layer, and an output layer. Here, the feature vectors are modelled (described in Section 3.2.1) as input vectors connected to each hidden neuron, where the hidden neurons are the distance between the centre and the inputs in the first stage. In the second stage, the connection weights between the hidden layer and the output layer are calculated in such a way that an error criterion such as the common Mean Squared Error (MSE) is minimized over all the data set. The typical RBF neural network is determined by the Eq. (10).

$$y = \sum_{i=1}^{N} w_i R_i(x) + b \tag{10}$$

where, w_i are weights for all training set y, b is bias, N number of the neurons in the hidden layer and R is the RBF activation function denoted in Eq. (11) as:

$$R_i(x) = \varphi(||x - c_i||) \tag{11}$$

where, φ is the radial basis function that present the non-linear feature of the model, c_i are the centers (obtained from X-means). The well-known RBF is given by the Gaussian function denoted in Eq. (12) as:

$$\varphi(r) = \exp\left(-r^2/2\sigma^2\right) \tag{12}$$

where, r is the Euclidean distance between the centroid c_i (obtained from X-means) and the input vector x, σ is the spread (widths) parameter, and simplicity variances are predefined as '1'. A pseudo code for the proposed AFDD employing RBF neural network (RBFNN) is presented in the Algorithm 2.

(2) **Probabilistic neural network:** The PNN is a feed-forward neural network comprising four layers: an input layer, a pattern layer, a summation layer, and an output layer [33]. It is mainly suitable for solving multi-class classification problems because it has fast processing time making it appropriate for real time implementation. The input layer consists of the feature vector combined with Gaussian functions in the pattern layer for each of the faulty and non-faulty classes. Each input node in the pattern layer calculates the value of the Gaussian function of the Euclidean distance between the given input vector and one training input vector determining the close proximity between the input vector and the training input vector. The pattern layer acts similar to the hidden layer in an RBF network [34]. The third summation layer adds the weighted hidden node values for each class and forms a probability density function (PDF) as its net output. Finally, the output layer performs a complete transfer function on the output of the second layer and selects the maximum of these probabilities, producing a '1' for that class and a '0' for the other classes to determine the associated class label. The PNN is determined by Eq. (13) as follows:

Algorithm 2 Pseudo code for proposed AFDD employing RBFNN.

Require: Temperature and power feature of TUs = $\{f_1, f_2, ..., f_{12}\}$
Class label = $\{C_0, C_1, ..., C_5\}$ $\Leftarrow$ *using the Eq.* (8) & (9)
RBF function = $\varphi(r)$, spread = σ, bias= b, weight matrix = w

Training Phase:
Step 1:
$N = Total\ no.\ of\ TU\ patterns$
$i = pattern\ no.$
for all $i = 1$ *to* N **do**
  for all $j = 1$ *to* 12 **do**
    $Employed\ values\ for\ centroid = \mu_0, \mu_1, ..., \mu_5$ $\Leftarrow$ *using Algorithm* 1
    $Calculate\ RBF\ function$ $\Leftarrow$ *using the Eq.* (11) & (12)
    $Determine\ hidden\ nodes\ values\ for\ hidden\ layer$
    $RBFNN\ trained\ \&\ generate\ final\ output\ classes$ $\Leftarrow$ *using the Eq.* (10)
  end for
end for
Testing Phase:
Step 2:
The trained RBFNN model applied on testing dataset for prediction on rest of TUs
Validation:
Step 3:
The actual and predicted target matrix has been compared
The performance has been validated $\Leftarrow$ *using the Eq.* (16), (17), & (18)

$$O_{sc} = \sum_{i=1}^{N_c} O_{pci} = \frac{1}{\sigma_c \sqrt{2\pi}} \sum_{i=1}^{N_c} \exp - \left(\frac{||s - t_{ci}||^2}{2\sigma_c^2} \right) \tag{13}$$

Where s is the input (TU feature) vector, t_{ci} is the i^{th} training input vector, N_c is the number of training input vectors in the c^{th} class. t_{ci} and σ_c represent the center and spread (or width) of the Gaussian bell curve associated with the c^{th} input class respectively. In addition, $\sigma_c\sqrt{2\pi}$ is determined by the width factor associated with the c^{th} class. The O_{pci} is close to 1 when s is near t_{ci}, and close to 0 when s is far from t_{ci} in a Euclidean space. Therefore, it can be concluded that the maximum PDF is obtained for the class, which yields the highest value of O_{sc}.

(3) **Generalized regression neural network:** The GRNN is a modified neural network of RBF and PNN neural networks [35]. The GRNN can provide a solution to any approximation problem by estimating the PDF for any type of data [36]. The main benefits of this network are that it does not require any repetitive training process making it advantageous with sparse data as the regression surface is defined instantly. This neural network also consists of four layers: an input layer, a pattern layer, a summation layer, and an output layer. The input layer is fully connected to the pattern layer and measures the distance between each of the neurons of these two layers. It uses the radial basis function as the activation function in this layer. Later it sums the two neurons in the summation layer: one neuron sums the weighted output of the pattern layer and other calculates unweighted outputs and divides them in the output layer to obtain a final output.

$$O_i = \frac{\sum_{i=1}^{n} y_i \exp[-D(x, x_i)]}{\sum_{i-1}^{n} \exp[-D(x, x_i)]} \tag{14}$$

where, the D is Gaussian function defined as,

$$D(x, x_i) = \sum_{k=1}^{m} \left(\frac{x_k - x_{ik}}{\sigma} \right)^2 \tag{15}$$

where, n is number of input nodes, y_i is weighted connection between the i^{th} neuron in the pattern layer and summation node; m is number of items in input vector; and x_k and x_{ik} are the j^{th} element of x and x_i, respectively. The σ is the spread (or width), determines the generalization performance of the GRNN. In general, a larger σ value may result in better generalization; its optimal value is determined through trial and error. In conventional GRNN applications, all units in the pattern layer have the same single spread. The performance of these algorithms is depending on the selection of the parameters and here the default parameters have been used for simplicity.

3.2.4. Statistical validation

The proposed model has been validated through established and well-known statistical measures. The accuracy, error rate, sensitivity, and specificity have been calculated to analyse the performance of the proposed work. The overall accuracy (A_c) of the model has been calculated measuring the number of TUs correctly classified from each case in the study experiment TUs. The performance metrics have been calculated measuring the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates. Also, the error rate (E_r) has been calculated from the miss-classified TUs. The mathematical interpretation of A_c is described as follows,

$$A_c = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

Sensitivity (S_n) and specificity (S_p) have been employed to quantify the performance of the two case studies and obtain detection and diagnosis results of the proposed framework. Sensitivity is the proportion of faulty and non-faulty TU behaviour correctly detected by the test, where specificity refers to the proportion of the TU behaviours that are correctly excluded from the group that they should not actually belong to. These analyses are vital to confirm that specific faulty and non-faulty behaviour can be properly implemented using the proposed framework towards addressing energy wastage and building portfolio inefficiencies.

$$S_n = \frac{TP}{TP + FN} \tag{17}$$

$$S_p = \frac{TN}{TN + FP} \tag{18}$$

4. Experimental result analysis

The proposed AFDD method has been investigated on two real building's HVAC TU datasets for their faulty and non-faulty behaviour analysis and to determine the potential savings that can be optimised through the proposed approach. These investigations have been performed on two real commercial buildings based in central of London, United Kingdom. Both building's information is provided first in this section and the obtained result discussed later executing/ performing the proposed system on these buildings. This study includes exploration, analysis, learning, and prediction on the TU data employing supervised ML algorithms. The process relies on the data properties, amount of data availability and different parameter values, thus this investigation is rigorous and time intense. One-day of TU data is considered as an individual pattern and is represented by twelve unique feature vectors that have been developed by the proposed feature engineering method (described in Section 3.2.1). The experiment has been conducted by separating the whole dataset into two phases: training and testing. The training and testing data have been varied by different percentages for the investigation. The training has been started, varying from 5% to 80%

and conversely testing began from 95% to 20% in decrements of 5%. This is a trial-and-error method to discover the optimal training and testing ratio that can achieve the best performance for this type of dataset. The buildings historic data considered in the time frame ranges from July 2015 to July 2018 for the case study 1 and January 2018 to July 2018 for case study 2. A description of both buildings has been detailed in Table 3.

Table 3: Investigated case study buildings physical information and experimental data details

Building details	**Case study-1**	**Case study-2**
Building type	Commercial	Commercial
Area of floor space (sq. ft.)	157,000	200,000
Building's height (meter)	73.42	30.5
Established year	1976	1982
Retrofit year	2008	2000
Number of floors	17^{th}	7^{th}
Total TUs	731	516
Operating TUs	723	490
Investigation starts	17^{th} July 2015	1^{st} January 2018
Investigation end	31^{st} July 2018	31^{st} July 2018
Investigated working days	769	147
Investigated working hours	12	12

The experiments have been carried out using Matlab R2018b tool on an Intel(R) Core(TM) i5 processor@ 3.30 GHz running Windows 7 Enterprise 64-bit operating system with a 7856-MB NVIDIA Graphics Processing Unit (GPU).

4.1. Result analysis: Case study-1

The experimental result analysis has been performed on three consecutive years of TU data comprising over half a million data points for case study 1. The results obtained from the proposed RBF framework over this case study are shown in Table 4. An accuracy of $0.978 \equiv 97.8\%$ has been achieved with 5% training data, however the highest training accuracy is achieved with 40% training data i.e., $0.996 \equiv 99.6\%$. While the highest testing accuracy is obtained with 60% dataset is $0.957 \equiv 95.7\%$. Although it has been seen for every training and testing set an average accuracy has been achieved

of more than 0.965 ≡ 96.5% and 0.891 ≡ 89.1% with the average error rate of 0.035 ≡ 3.5% and 0.978 ≡ 97.8% respectively. Also, the sensitivity and specificity have been employed to quantify the performance of the algorithm's results for identification of different TU behaviour. Often, the sensitivity and specificity of an experiment are inversely related and the selection of the optimal balance between the sensitivity and specificity depends on the purpose of the test. Here, the sensitivity 0.997 ≡ 99.7% and specificity 0.998 ≡ 99.8% are achieved with 40% TU data. For testing, the highest sensitivity 0.988 ≡ 98.8% and specificity 0.996 ≡ 99.6% are achieved with 60% TU data, specificity and sensitivity for both the training and testing resulting in good scores. This depicts that the RBF neural network can classify each of the classes appropriately with very little error. By this method, the faulty and non-faulty classes are truly identified by means of their activities. Thus, the TUs which are misbehaving are recognised accurately and correctly classified to their corresponding class. The RBF performance shows the effect of the method over-fitting after 40% training data, thus accuracy does not improve with increased training. The RBF approach needs less time to train a model as no repetition is required to reach the optimum parameters.

Table 4: Training and testing results using RBF for case study-1

Performance of RBF											
Training Phase	Data	A_c	E_r	S_n	S_p	Testing Phase	Data	A_c	E_r	S_n	S_p
	5%	0.978	0.022	0.997	0.996		95%	0.923	0.077	0.973	0.979
	10%	0.976	0.024	0.995	0.998		90%	0.915	0.085	0.982	0.988
	15%	0.965	0.035	0.995	0.998		85%	0.899	0.101	0.986	0.992
	20%	0.985	0.015	0.993	0.998		80%	0.933	0.067	0.988	0.993
	25%	0.984	0.016	0.995	0.998		75%	0.937	0.063	0.991	0.994
	30%	0.991	0.009	0.993	0.998		70%	0.936	0.064	0.988	0.993
	35%	0.992	0.008	0.992	0.998		65%	0.935	0.065	0.989	0.994
	40%	**0.996**	0.004	0.997	0.998		60%	**0.957**	0.043	0.988	0.996
	45%	0.993	0.007	0.989	0.998		55%	0.933	0.067	0.985	0.996
	50%	0.994	0.006	0.992	0.998		50%	0.924	0.076	0.989	0.997
	55%	0.992	0.008	0.99	0.998		45%	0.912	0.088	0.988	0.996
	60%	0.992	0.008	0.991	0.998		40%	0.911	0.089	0.989	0.995
	65%	0.988	0.012	0.991	0.998		35%	0.904	0.096	0.988	0.995
	70%	0.987	0.013	0.991	0.992		30%	0.899	0.101	0.99	0.996
	75%	0.983	0.017	0.991	0.998		25%	0.898	0.102	0.989	0.996
	80%	0.982	0.018	0.989	0.998		20%	0.891	0.109	0.986	0.996

The obtained RBF results have been compared with two other neural networks: PNN and general GRNN. Both are based on the RBF function and widely used for classification purposes. PNN differentiates from RBF and works on estimating the probability density function while RBF works on an iterative function approximation. The obtained results from PNN are tabulated in Table 5. In the PNN case, the highest training classification accuracy of 0.815 ≡ 81.5% was achieved with 40% data and testing accuracy 0.815 ≡ 81.5% achieved with 65% data. Thus, in terms of accuracy, PNN has not achieved comparative results to RBF for the faulty and non-faulty TU classification. While an average accuracy is obtained for both the training and testing

of approximately $0.81 \equiv 81\%$, it is 18% below the RBF performance. Likewise, the sensitivity and specificity obtained was $0.777 \equiv 77.7\%$ and $0.996 \equiv 99.6\%$ for training and $0.783 \equiv 78.3\%$ and $0.996 \equiv 99.6\%$ for testing. Though, the specificity is high and similar to RBF results, PNN can be seen to detect the TUs which do not truly belong to the particular class, but does not predict the right class setting. Also, the TUs that do not actually belong to specific classes are detected properly, affecting the sensitivity outcomes for this network. This mainly happens due to the similar types of feature values. As the features are generally formed by calculating the area of temperature and power curve, if the total area value of one TU is similar to another TU, they could be classified into the same group. But the TU may have different behaviour in terms of power demand aspects or control strategy aspects. Thus, it is crucial to choose a classifier that can deal with data non-linearity and the feature overlapping tendencies. However, it is observed that the training and testing performance are similar for various data percentages, inferring learning with the PNN networks are steady.

Table 5: Traning and testing results using PNN for case study-1

Performance of PNN											
Training Phase	Data	A_c	E_r	S_n	S_p	Testing Phase	Data	A_c	E_r	S_n	S_p
	5%	0.812	0.188	0.799	0.996		95%	0.814	0.186	0.787	0.997
	10%	0.811	0.189	0.757	0.997		90%	0.805	0.195	0.753	0.997
	15%	0.81	0.19	0.763	0.997		85%	0.807	0.193	0.765	0.996
	20%	0.812	0.188	0.774	0.996		80%	0.813	0.187	0.784	0.996
	25%	0.81	0.19	0.765	0.996		75%	0.808	0.192	0.766	0.995
	30%	0.811	0.189	0.772	0.996		70%	0.808	0.192	0.764	0.996
	35%	0.81	0.19	0.786	0.996		65%	**0.815**	0.185	0.783	0.996
	40%	**0.815**	0.185	0.777	0.996		60%	0.811	0.189	0.773	0.996
	45%	0.814	0.186	0.782	0.996		55%	0.814	0.186	0.783	0.996
	50%	0.812	0.188	0.772	0.996		50%	0.809	0.191	0.769	0.996
	55%	0.814	0.186	0.785	0.996		45%	0.813	0.187	0.786	0.992
	60%	0.812	0.188	0.775	0.996		40%	0.809	0.191	0.774	0.992
	65%	0.812	0.188	0.776	0.996		35%	0.811	0.189	0.777	0.995
	70%	0.814	0.186	0.785	0.996		30%	0.814	0.186	0.788	0.997
	75%	0.814	0.186	0.786	0.995		25%	0.814	0.186	0.787	0.995
	80%	0.815	0.185	0.785	0.995		20%	0.813	0.187	0.789	0.995

Unlike PNN, GRNN is also a variant of RBF neural network and has similar architectures, except there is a fundamental difference; GRNN performs regression where the target variable is continuous, whereas PNN performs classification where the target variable is categorical. Both have fixed smoothing parameters. The accuracy achieved is very low compared to RBF and PNN, with results tabulated in Table 6. The maximum accuracy of $0.491 \equiv 49.1\%$ is achieved with 40% training data set and $0.493 \equiv 49.3\%$ with 55% testing dataset, along with very high error rates of $0.509 \equiv 50.9\%$ and $0.507 \equiv 50.7\%$ for training and testing cases, respectively. Sensitivity is very low <0.05 depicting that this GRNN is unable identify actual TU behaviour failing to predict the corresponding class of each TU pattern. The specificity is high, on average $0.99 \equiv 99\%$ and $0.98 \equiv 98\%$ for training and testing dataset which is quite similar to the other

networks employed here. It reflects the true negative rate is high, thus the misclassified patterns of each class are truly not belonging to that class. Thus, the GRNN can capture the true negative patterns but fails to detect positive patterns. According to statistical distribution theory, event-based data for faulty and non-faulty analysis may have a certain overlap in terms of their probability distributions, which may lead to contradictions between sensitivity and specificity.

Table 6: Traning and testing results using GRNN for case study-1

Performance of GRNN											
Training Phase	Data	A_c	E_r	S_n	S_p	Testing Phase	Data	A_c	E_r	S_n	S_p
	5%	0.458	0.542	0.053	0.992		95%	0.464	0.536	0.057	0.985
	10%	0.482	0.518	0.051	0.998		90%	0.476	0.524	0.051	0.997
	15%	0.485	0.515	0.045	0.989		85%	0.482	0.518	0.056	0.998
	20%	0.465	0.535	0.052	0.998		80%	0.463	0.537	0.054	0.998
	25%	0.496	0.504	0.055	0.99		75%	0.492	0.508	0.056	0.987
	30%	0.467	0.533	0.055	0.995		70%	0.466	0.534	0.001	0.996
	35%	0.461	0.539	0.043	0.997		65%	0.463	0.537	0.042	0.993
	40%	**0.491**	0.509	0.044	0.995		60%	0.491	0.509	0.046	0.958
	45%	0.491	0.509	0.05	0.992		55%	**0.493**	0.507	0.049	0.933
	50%	0.485	0.515	0.051	0.999		50%	0.481	0.519	0.039	0.998
	55%	0.478	0.522	0.043	0.996		45%	0.483	0.517	0.043	0.966
	60%	0.484	0.516	0.042	0.996		40%	0.479	0.521	0.041	0.969
	65%	0.482	0.518	0.002	0.997		35%	0.482	0.518	0.003	0.965
	70%	0.483	0.517	0.002	0.995		30%	0.485	0.515	0.003	0.996
	75%	0.489	0.511	0.042	0.997		25%	0.493	0.507	0.043	0.968
	80%	0.490	0.510	0.026	0.998		20%	0.491	0.509	0.024	0.969

4.2. Result analysis: Case study-2

The experimental result analysis has been performed over seventy thousand TU data in case study 2. This study considered seven months of data from January 2018 to July 2018. The investigated TUs for this building are found to be mostly badly behaved and this is supported by the expert building engineers who viewed the data. The misbehaving TUs belong to the three classes (C_3 to C_5) with high control temperature and power demands. But, among these three classes, some of the TUs that are not displaying similar behaviour to the group mislead the classification results. Here, the RBF results for case study 2 are tabulated in Table 7 and the highest values are highlighted in bold. The highest accuracy for this network achieved in 40% training case is $0.944 \equiv 94.4\%$ and $0.931 \equiv 93.1\%$ achieved for the 60% testing case. In the case of training, the sensitivity and specificity are $0.953 \equiv 95.3\%$ and $0.994 \equiv 99.4\%$; whereas for testing $0.779 \equiv 77.9\%$ and $0.951 \equiv 95.1\%$ respectively. For training both specificity and sensitivity are high; thus, the true positive and true negative rate are good for RBF. However, for testing, the sensitivity was quite low, which depicts the TUs that actually belong to each class can be detected with the training data, but it is less able to detect that they truly belong to that class for testing while the training TU data is not available.

Table 7: Traning and testing results using RBF for case study-2

Performance of RBF											
Training Phase	Data	A_c	E_r	S_n	S_p	Testing Phase	Data	A_c	E_r	S_n	S_p
	5%	0.919	0.081	0.996	0.985		95%	0.765	0.235	0.698	0.985
	10%	0.917	0.083	0.952	0.982		90%	0.836	0.164	0.723	0.982
	15%	0.927	0.073	0.937	0.979		85%	0.862	0.138	0.766	0.916
	20%	0.926	0.074	0.921	0.994		80%	0.88	0.12	0.754	0.951
	25%	0.935	0.065	0.911	0.993		75%	0.893	0.107	0.769	0.955
	30%	0.935	0.065	0.972	0.994		70%	0.901	0.099	0.723	0.944
	35%	0.925	0.075	0.962	0.998		65%	0.908	0.092	0.752	0.939
	40%	**0.944**	0.056	0.953	0.994		60%	**0.931**	0.069	0.779	0.951
	45%	0.939	0.061	0.973	0.991		55%	0.915	0.085	0.744	0.953
	50%	0.934	0.066	0.992	0.997		50%	0.919	0.081	0.759	0.907
	55%	0.923	0.077	0.995	0.987		45%	0.921	0.079	0.765	0.964
	60%	0.923	0.077	0.992	0.993		40%	0.924	0.076	0.754	0.959
	65%	0.933	0.067	0.991	0.998		35%	0.926	0.074	0.762	0.917
	70%	0.922	0.078	0.984	0.985		30%	0.928	0.072	0.793	0.962
	75%	0.912	0.088	0.941	0.987		25%	0.929	0.071	0.807	0.951
	80%	0.913	0.087	0.995	0.982		20%	0.913	0.087	0.789	0.915

The results obtained from PNN are tabulated in Table 8. From the PNN outcomes, it has been found that the maximum accuracy obtained with 50% training is $0.651 \equiv 65.1\%$ and 60% testing is $0.642 \equiv 64.2\%$. The accuracy for this case study was nearly 20% lower accuracy than case study 1. Although both buildings have the same parameters and feature representation, their TU behaving nature are physically different and that impacts the classification results. This building's TUs are quite badly behaved TUs and demonstrate many malfunctions, also sensor locations are not appropriately set to collect the correct discharge air temperature from the nearest unit, greatly affecting the TUs working conditions. The resultant sensitivity is lower as well at 0.198 for 50% training and 0.158 for 60% testing set. Therefore, the true detection of each class is not accurately predicted. While the specificity is high it illustrates the TUs are not truly predicted to their corresponding classes.

The GRNN results are tabulated in Table 9, and it is observed that unlike the previous case study results, here also the GRNN fails to classify the TU into their corresponding faulty and non-faulty classes. However, one difference is noticed,, maximum accuracy is achieved with lower level of training data; 25% of training data obtained accuracy $0.498 \equiv 49.8\%$ and for testing accuracy obtained $0.495 \equiv 49.5\%$ with 60% TU dataset. Then sensitivity is also found very low <0.05 thus GRNN fails to identify the actual TU behaviour and to predict the corresponding class labels. Conversely, the specificity is high, on average $0.70 \equiv 70\%$ and $0.93 \equiv 93\%$ for training and testing dataset. Thus, the true negative rate is higher for testing than training; the patterns of each class that are misclassified truly do not belong to that class. Consequently, the GRNN can capture the true negative patterns but fails to detect positive patterns. Overall, for the building in case study 2 the sensitivity achieves low, and specificity achieves high.

Table 8: Traning and testing results using PNN for case study-2

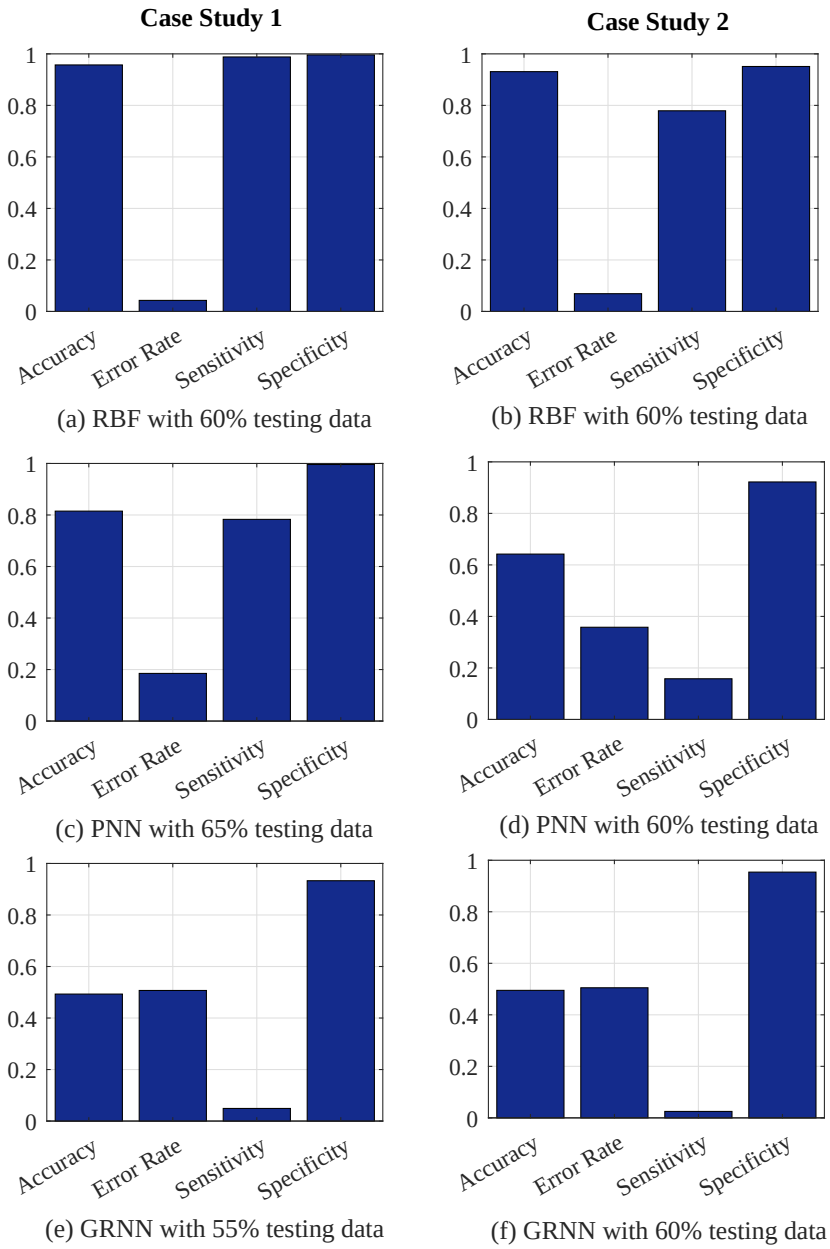
Performance of PNN											
Training Phase	Data	A_c	E_r	S_n	S_p	Testing Phase	Data	A_c	E_r	S_n	S_p
	5%	0.647	0.353	0.252	0.885		95%	0.636	0.364	0.182	0.938
	10%	0.648	0.352	0.213	0.885		90%	0.638	0.362	0.153	0.976
	15%	0.643	0.357	0.212	0.787		85%	0.636	0.364	0.156	0.912
	20%	0.643	0.357	0.202	0.889		80%	0.64	0.36	0.159	0.983
	25%	0.644	0.356	0.201	0.885		75%	0.636	0.364	0.156	0.921
	30%	0.643	0.357	0.201	0.826		70%	0.637	0.363	0.164	0.992
	35%	0.642	0.358	0.187	0.877		65%	0.641	0.359	0.159	0.917
	40%	0.641	0.359	0.207	0.885		60%	**0.642**	0.358	0.158	0.922
	45%	0.642	0.358	0.185	0.788		55%	0.637	0.363	0.162	0.913
	50%	**0.651**	0.349	0.198	0.749		50%	0.638	0.362	0.159	0.927
	55%	0.644	0.356	0.2	0.897		45%	0.639	0.361	0.161	0.931
	60%	0.643	0.357	0.189	0.899		40%	0.638	0.362	0.163	0.927
	65%	0.644	0.356	0.198	0.745		35%	0.639	0.361	0.164	0.932
	70%	0.6428	0.3572	0.203	0.698		30%	0.641	0.359	0.157	0.924
	75%	0.643	0.357	0.202	0.882		25%	0.638	0.362	0.154	0.935
	80%	0.642	0.358	0.187	0.974		20%	0.639	0.361	0.155	0.944

Table 9: Traning and testing results using GRNN for case study-2

Performance of GRNN											
Training Phase	Data	A_c	E_r	S_n	S_p	Testing Phase	Data	A_c	E_r	S_n	S_p
	5%	0.491	0.509	0.068	0.755		95%	0.493	0.507	0.025	0.912
	10%	0.493	0.507	0.057	0.635		90%	0.493	0.507	0.021	0.923
	15%	0.497	0.503	0.059	0.695		85%	0.494	0.506	0.022	0.945
	20%	0.492	0.508	0.063	0.715		80%	0.494	0.506	0.024	0.932
	25%	**0.498**	0.502	0.053	0.732		75%	0.492	0.508	0.024	0.957
	30%	0.497	0.503	0.054	0.758		70%	0.493	0.507	0.027	0.941
	35%	0.493	0.507	0.053	0.745		65%	0.494	0.506	0.024	0.943
	40%	0.493	0.507	0.051	0.755		60%	**0.495**	0.505	0.025	0.954
	45%	0.496	0.504	0.043	0.751		55%	0.493	0.507	0.028	0.944
	50%	0.495	0.505	0.049	0.732		50%	0.493	0.507	0.026	0.954
	55%	0.494	0.506	0.053	0.741		45%	0.495	0.505	0.026	0.938
	60%	0.494	0.506	0.05	0.775		40%	0.494	0.506	0.026	0.956
	65%	0.493	0.507	0.05	0.753		35%	0.495	0.505	0.029	0.945
	70%	0.495	0.505	0.052	0.725		30%	0.495	0.505	0.027	0.938
	75%	0.496	0.504	0.051	0.712		25%	0.494	0.506	0.029	0.935
	80%	0.493	0.507	0.042	0.699		20%	0.495	0.505	0.027	0.937

4.3. Performance Comparison:

From the result analysis, it has been found that the x-RBF achieved good results among the compared methods for both the case studies. The results are compared by varying different training and testing percentages and also by statistical validation metrics. Maximum performance score were obtained with 40% training set and with 60% testing set. Thus, based on these statistical scores, the proposed x-RBF have been considered an ideal method to perform the AFDD for these case study buildings. The scores have been plotted as a bar graph in Fig. 7.



(a) RBF with 60% testing data

(b) RBF with 60% testing data

(c) PNN with 65% testing data

(d) PNN with 60% testing data

(e) GRNN with 55% testing data

(f) GRNN with 60% testing data

Figure 7: Testing performance comparison between different experimented methods for both the case studies

After investigating the three neural networks, x-RBF, PNN, GRNN for both case studies, the notable features that have significant effect on the performance of each method have been compared and are tabulated in Table 10. The performance of the NN depends on the number of inputs (feature vector for the energy data) that are same twelve-dimensional feature vector for all the methods. The NN function approximation values have been chosen here by default to maintain low computational

complexity. The amount of training data that achieves the optimal performance for the TU fault identification is also given. The faulty and non-faulty TU detection and misidentification have been considered here. It has been found that the true positive rate is high for the x-RBF where the faulty TU has been identified truly and the false negative is appreciable where non-faulty TUs are identified properly along with low error rate. The PNN and GRNN do perform well with small datasets but with larger datasets, they become impractical.

Table 10: A table for performance comparison between experimented NNs.

Parameters	Case study 1			Case study 2		
	x-RBF	PNN	GRNN	x-RBF	PNN	GRNN
No. of input neurons	12	12	12	12	12	12
No. of NN layers	3	4	4	3	4	4
Seeding function approximation value	1.0	0.1	1.0	1.0	0.1	1.0
No. of bias required in the NN layers	Both layer	First layer	First layer	Both layer	First layer	First layer
Data required for optimal training	40%	40%	40%	40%	50%	25%
Faulty TU detection performance	0.988	0.773	0.046	0.779	0.158	0.025
Non-faulty TU detection performance	0.996	0.996	0.958	0.951	0.917	0.954
Error	0.043	0.189	0.509	0.069	0.359	0.505
Time complexity	Low	Medium	Medium	Low	Medium	Medium
Building energy saving capacity	High	Medium	Low	High	Medium	Low

5. Impact on energy optimisation

From the result analysis of the proposed framework, it has been found that RBF has superior performance compared to the NN methods employed for these two real TU data case study buildings in London. The advantage of the proposed model is it can identify faulty HVAC TUs based on their control temperature and energy consumption, irrespective of any particular fault type. So that, the proposed research would have real effect and immediate and long term impact on long term BEMS energy optimisation by early fault prediction. In more detailed discussion, it has been found during previous research by the authors [23], that non-faulty TUs usually consume energy between 0.1 to 0.2 kWh for both heating and cooling phenomenon. Whereas, is the case of faulty TUs, the energy consumption reaches a maximum 1 kWh (as shown in Fig. 8). Thus, anticipation of the faults would instil enormous energy consumption reductions as well as cost benefits. This can be realised from the Eq. (19).

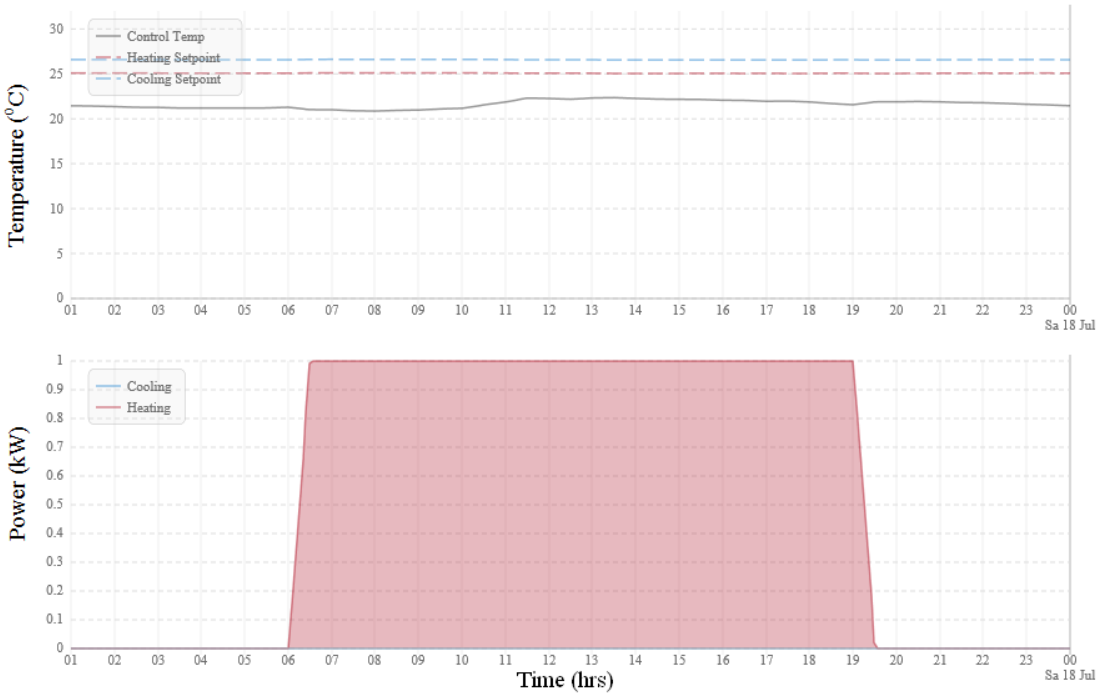


Figure 8: The example of faulty heating TU from the case study-1

$$E = P \times t \times n \tag{19}$$

Where, E, is the electrical energy, P is the power consumption of each unit, t is the operational hours/ running time of each unit inside the building, n is the total number of faulty TUs. Subsequently, the average charges for each unit are 14.53 pence/kWh for London region [37]. This according to that the costing for per day as follows, where, C is the total cost of the units, R, is the average rate/ pence.

$$C = R \times E \tag{20}$$

In the case of the first building (case study-1), the proposed collaborative ML framework achieved an average sensitivity and specificity of 98.8% and 99.6% respectively in the testing phase. Here, sensitivity and specificity demonstrated the ability to identify faulty and non-faulty units respectively. In one day (18^{th} July 2015) the proposed model detected 124 faulty TUs where the building engineers confirmed the number of faulty units was 125 (among 731 TUs, results 17.28% faults). If the power consumption of a faulty unit is x kWh (as this figure varies with other parameters) and each unit operates 12hrs/ day then, total amount of energy consumption would be $E = (x \times 12 \times 124)kWh = 1488xkWh$ that can be saved by the proposed work. Subsequently, this much of energy costs $C = (1488x \times 14.53pence) =\sim$ £220 savings can be made for the first building in just one day on these fault alone.

In the case of the second building (case study-2), the proposed work achieved average sensitivity and specificity of 77.90% and 95.1% respectively. On a particular day (2^{nd} January 2018), the building engineer confirmed 180 faulty TUs (results 36.73% faults) where the proposed work identified 140 number of faults. Hence, the amount of energy of second building would have safeguarded is $E = (x \times 12 \times 140)kWh = 1680xkWh$ which costs approximately $C = (1680x \times 14.53pence) =\sim £250$ in one day for the second building again on these faults alone.

Therefore, the research establishes a long-term, implementable solution for energy efficiency implementation and cost optimisation in real buildings. The research considers automatic fault detection, which needs to be followed by the building engineers for diagnosis purposes. The detection provides the TU 'ID' number and fault type to the building engineers where the associated faults are then managed by experts. These faults include the problems of fan, motor, belt, pulley, variable speed drive, sensors etc. This supportive assistance would have a major impact on standards creation for building BMS fault detection and rectification as well as highlighting the most common faults observed, enabling HVAC manufacturers and building designers to take pre-emptive actions in building creation and retrofitting.

6. Discussions

In building management optimising energy usage and maintaining optimum building performance in real-world scenarios is very challenging. Researchers have dedicated enormous efforts to develop proficient approaches to deal with the real challenges in building sectors to improve performance and reduce energy waste, mainly at the component level via building automation systems. However, many problems are left unobserved and/or unsolved because of the vast data and the complex nature of building management systems (BMSs). A buildings BMS produces vast amounts of data which mostly goes misunderstood and rearley analysed due to the lack of available expert building engineers, overhead costs, as well as huge time complexity. Thus, user friendly data mining and machine learning methods are drawing attention for BMS data analysis and building support, however this is a complex route to discover effective methods as knowledge discovery and learning are highly data dependent, where every data signifies a specific behaviour. This makes the field extremely motivating and ripe formore focused exploration. Pivotally, among numerous units used in buildings, the fan coil unit (FCU) has not been explored or compared to other major units such as AHU, chillers, and boilers, yet this is the unit closest to the end user. Equipment failure and performance degradation of HVAC TUs often go ignored until they cause a noticeable impact on occupant's thermal comfort, trigger an equipment-level alarm, deteriorate equipment lifespan, noted excessive energy consumption or costly breakdowns. Therefore, early detection of unexpected behaviour and subsequent remedial best practice can assist with these issues effortlessly. Here, the AFDD study began by analysing TU performance to recognise process problems in, e.g. temperature, flows, pressures, level, power, control signals, etc. Specifically, it is an examination of the device, which is

not working desirably and unable to maintain a 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 occupant discomfort and significant amount of energy loss.

This paper presents a data-driven based framework for automatic TU fault identification and associated energy conversion. This method combines new feature engineering with a modified RBF (x-RBF) neural network followed by energy estimation calculation. While analysing the real TU case study data, it has been found that the non-parametric NN model is suitable for non-linear problems. Its learning and classifying capability are quite impressive over other compared NN models. The training accuracy for both distinct case studies are above $\sim$99% and $\sim$94%. This defines the model's suitability for identifying the specific faulty TUss. Subsequently, the correct/true identification of faulty and non-faulty patterns could provide better opportunities to identify and attend to problems when they occur, with large savings, both financial and in energy efficiencies to be made. Only one day energy conservation calculation has been performed here and a significant saving has been observed. Thus, this research work demonstrates that the implementation of such approaches in BMS could lead to large scale efficiencies in building management, design behaviour.

7. Conclusions and future work

This study provides a fault identification and energy saving estimation framework using the proposed x-RBF method. This proposed method is a combination of pseudo labelling with X-means clustering and RBF neural network to auto-classify the 'faulty' and 'non-faulty' behaviour to assist building managers in fast detect and diagnosis issues inside building. A new feature engineering method has been proposed to represent the high dimensional TU data into a meaningful lower dimensional data stream. The feature engineering method is performed by discovering events from the time series energy data using the PID controller response curve. The investigation began by analysing historic data from a small but pivotal HVAC sub-component (i.e., TU). The algorithm concentrates on proposing data-driven based approaches, which could eventually classify faults from all different units (fan coil unit, variable air volume, air handling unit, chiller, boiler, etc.) regardless of specific faults. The approach is aimed to not only detect and diagnose equipment failures, but also provide significant energy savings and well-being enhancements through pre-emptive maintenance, behaviour analysis and predictive building identification. The proposed method has been employed to the two real-life case study buildings in London and found x-RBF outperforms over other compared methods. The x-RBF has a strong tolerance to outliers and performs efficient training processes employing the universal approximation property which resulted in significantly lower-error estimation both in the training and testing stages. From the experimental results it has been observed that the overall performance of x-RBF can be applied successfully for the detection of terminal unit

faults for these buildings. A future extension will evaluate the performances of hybrid models with energy data from other type of HVAC unit.

Acknowledgement

This research project was funded by Innovate UK (EP/M506734/1). The authors are grateful to the Demand Logic team for providing us the data and feedback on building services and related performance issues.

References

[1] Y. Zhao, T. Li, X. Zhang, C. Zhang, Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future, Renewable and Sustainable Energy Reviews 109 (2019) 85–101.

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

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

[4] Y. Yu, D. Woradechjumroen, D. Yu, A review of fault detection and diagnosis methodologies on air-handling units, Energy and Buildings 82 (2014) 550–562.

[5] W. Kim, S. Katipamula, A review of fault detection and diagnostics methods for building systems, Science and Technology for the Built Environment 24 (1) (2018) 3–21.

[6] B. Gunay, W. Shen, C. Yang, Characterization of a building's operation using automation data: A review and case study, Building and Environment 118 (2017) 196–210.

[7] P. De Wilde, Building performance analysis, John Wiley & Sons, 2018.

[8] F. Magoulès, H.-x. Zhao, D. Elizondo, Development of an rdp neural network for building energy consumption fault detection and diagnosis, Energy and Buildings 62 (2013) 133–138.

[9] X. Yu, S. Ergan, G. Dedemen, A data-driven approach to extract operational signatures of hvac systems and analyze impact on electricity consumption, Applied Energy 253 (2019) 113497.

[10] C. Fan, F. Xiao, C. Yan, C. Liu, Z. Li, J. Wang, A novel methodology to explain and evaluate data-driven building energy performance models based on interpretable machine learning, Applied Energy 235 (2019) 1551–1560.

[11] D. Li, G. Hu, C. J. Spanos, A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis, Energy and Buildings 128 (2016) 519–529.

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

[13] Y. Zhao, S. Wang, F. Xiao, Pattern recognition-based chillers fault detection method using support vector data description (SVDD), Applied Energy 112 (2013) 1041–1048.

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

[15] C. Shang, F. You, A data-driven robust optimization approach to scenario-based stochastic model predictive control, Journal of Process Control 75 (2019) 24–39.

[16] A. J. Sonta, P. E. Simmons, R. K. Jain, Understanding building occupant activities at scale: An integrated knowledge-based and data-driven approach, Advanced Engineering Informatics 37 (2018) 1–13.

[17] S. Vaghefi, M. Jafari, J. Zhu, J. Brouwer, 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) (2016) 600–610.

[18] A. Rogers, F. Guo, B. Rasmussen, A review of fault detection and diagnosis methods for residential air conditioning systems, Building and Environment 161 (2019) 106236.

[19] K. Yan, A. Chong, Y. Mo, Generative adversarial network for fault detection diagnosis of chillers, Building and Environment (2020) 106698.

[20] X. Luo, K. Fong, Y. Sun, M. Leung, Development of clustering-based sensor fault detection and diagnosis strategy for chilled water system, Energy and Buildings 186 (2019) 17–36.

[21] H.-x. Zhao, F. Magoulès, A review on the prediction of building energy consumption, Renewable and Sustainable Energy Reviews 16 (6) (2012) 3586–3592.

[22] 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 108 (2020) 950 – 966. doi:https://doi.org/10.1016/j.future.2018.02.019.

[23] M. Dey, M. Gupta, S. P. Rana, M. Turkey, S. Dudley, A pid inspired feature extraction method for hvac terminal units, in: Technologies for Sustainability (SusTech), 2017 IEEE Conference on, IEEE, 2017, pp. 1–5.

[24] M. Dey, S. P. Rana, S. Dudley, A case study based approach for remote fault detection using multi-level machine learning in a smart building, Smart Cities 3 (2) (2020) 401–419.

[25] K. J. Åström, T. Hägglund, Advanced PID control, ISA-The Instrumentation, Systems and Automation Society, 2006.

[26] P. Berkhin, A survey of clustering data mining techniques, in: Grouping multidimensional data, Springer, 2006, pp. 25–71.

[27] M. E. Celebi, K. Aydin, Unsupervised learning algorithms, Vol. 9, Springer, 2016.

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

[29] M. Dey, M. Gupta, M. Turkey, 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, 2017.

[30] S. Haykin, Neural networks: a comprehensive foundation, Prentice Hall PTR, 1994.

[31] M. Kubat, Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7., The Knowledge Engineering Review 13 (4) (1999) 409–412.

[32] H. Yu, T. Xie, S. Paszczyñski, B. M. Wilamowski, Advantages of radial basis function networks for dynamic system design, IEEE Transactions on Industrial Electronics 58 (12) (2011) 5438–5450.

[33] D. F. Specht, Probabilistic neural networks, Neural networks 3 (1) (1990) 109–118.

[34] H. Adeli, A. Panakkat, A probabilistic neural network for earthquake magnitude prediction, Neural networks 22 (7) (2009) 1018–1024.

[35] D. F. Specht, A general regression neural network, IEEE transactions on neural networks 2 (6) (1991) 568–576.

[36] S. Heddam, A. Bermad, N. Dechemi, Applications of radial-basis function and generalized regression neural networks for modeling of coagulant dosage in a drinking water-treatment plant: comparative study, Journal of Environmental Engineering 137 (12) (2011) 1209–1214.

[37] https://www.ukpower.co.uk/home_energy/tariffs_per_unit_kwh.