TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technology

Department of Software Science

Molika Meas 201436IVSM

XAI-based Fault Detection, Diagnosis and Monitoring Method for Air Handling Units

Master's Thesis

Supervisor

Juri Belikov

PhD

Co-supervisor

Ahmet Köse

PhD

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia Teaduskond
Tarkvaraarenduse Instituut

Molika Meas 201436IVSM

XAI BAASIL VEATUVASTUSE, DIAGNOSTIKA JA JÄLGIMISE MEETOD VENTILATSIOONIMASINATE JAOKS

Magistritöö

Juhendaja
Juri Belikov
PhD
Kaasjuhendaja
Ahmet Köse
PhD

Tallinn 2022

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author:	Molika Meas
Date:	May 10, 2022

Abstract

Fault detection and diagnosis (FDD) methods are designed to determine whether the equipment in buildings is functioning under normal or faulty conditions and aim to identify the type or nature of a fault. Recent years have witnessed an increased interest towards application of machine learning algorithms for FDD problems. Nevertheless, a possible problem is that users may find it difficult to understand the prediction process made by a black-box system that lacks interpretability.

This work presents a method that explains the outputs of an Extreme Gradient Boosting (XGBoost)-based classifier, using an Explainable Artificial Intelligence (XAI) technique. The method could benefit expert end-users requiring justification for the output made by the classifier. The method operates as follows: first, the XGBoost algorithm is used to detect and classify potential faults in the heating and cooling coil valves, sensors, and heat recovery of an air handling unit (AHU). Then, a XAI-based SHAP technique is used to provide the explanations, with a focus on the end-users, who, in this case, are HVAC engineers. To keep the explanations focused, we only show the user-selected sets of features and features with high attribution scores. We use sliding-windows to visualize the short history of the relevant features and to provide explanations for the diagnosed faults in the observed time period. This aims to provide information not only about what occurred at the time of fault presence but also about how the fault developed. Finally, the resulting explanations are assessed by seven HVAC engineers who are currently working in the field. The proposed approach is validated using real data collected from a shopping mall.

The thesis is in English and contains 60 pages of text, 8 chapters, 22 figures, and 6 tables.

Annotatsioon

Rikke tuvastamise ja diagnostika meetodite eesmärk on kindlaks teha, kas hoonete seadmed töötavad normaalsetes või vigastes tingimustes, ning nende eesmärk on kindlaks teha rikke tüüp või olemus. Viimastel aastatel on suurenenud huvi masinõppe algoritmide rakendamise vastu rikke tuvastamise ja diagnostika probleemide lahendamiseks. Võimalik probleem on siiski see, et kasutajatel võib olla raske mõista masinõppe mudeli poolt tehtud ennustusi, millel puudub tõlgendatavus.

Käesolevas töös esitatakse meetod, mis selgitab Extreme Gradient Boosting (XGBoost)-põhise klassifikaatori väljundeid, kasutades seletatava tehisintellekti (Explainable Artificial Intelligence, XAI) tehnikat. Meetodist võiksid kasu saada asjatundlikud lõppkasutajad, kes vajavad klassifikaatori tehtud väljundite põhjendamist. Meetod toimib järgmiselt: kõigepealt kasutatakse XGBoost-algoritmi, et tuvastada ja klassifitseerida võimalikke vigu kütte- ja jahutusventiilides, andurites ja õhukäitlusseadme soojustagastuses. Seejärel kasutatakse XAI-põhist SHAP-tehnikat selgituste andmiseks, keskendudes lõppkasutajatele, kes antud juhul on HVAC-insenerid. Visualiseerisime aegridade libisevate akende selgitusi, et anda teavet mitte ainult selle kohta, mis toimub vea esinemise ajal, vaid ka selle kohta, kuidas viga tekkis. Et hoida selgitused asjakohased, näitame ainult kasutaja valitud tunnuste ja kõrge SHAP-väärtusega tunnuste kogumeid. Lõpuks palusime leitud selgitusi hinnata HVAC-inseneridel. Väljapakutud lähenemisviis on valideeritud, kasutades kaubanduskeskusest kogutud andmeid.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 60 leheküljel, 8 peatükki, 22 joonist, 6 tabelit.

List of abbreviations and terms

1D-CNN	1 Dimentional Neural Network
AAT	Ambient Temperature
ACCVO	Cooling Coil Valve Opening
AHCVO	Heating Coil Valve Opening
AHRS	Heat Recovery Rotation Speed
AHRST	Supply Air Temperature after Heat Recovery
AHU	Air Handling Unit
AI	Artificial Intelligence
APAR	AHU Performance Assessment Rules
API	Application Programming Interface
ARAT	Return Air Temperature
ASAT	Supply Air Temperature
ASATCSP	Supply Air Temperature Calculated Setpoint
ASFPE	Supply Fan Static Pressure
ASFS	Supply Fan Speed
BMS	Building Management System
CIU	Contextual Importance and Utility
CNN	Convolutional Neural Network
DARPA	Defense Advanced Research Projects Agency
FDD	Fault Detection and Diagnosis
FN	False Negative
FP	False Positive
GDPR	General Data Protection Regulation
GradCAM	Gradient-weighted Class Activation Mapping
HVAC	Heating, Ventilation and Air Conditioning
LIME	Local Interpretable Model-agnostic Explanations
ML	Machine Learning
MLP	Multilayer Perceptron
NN	Neural Network
nZEB	near Zero Energy Building
PC	Principle Component

PCA	Principle Component Analysis
PID	Proportional–Integral–Derivative
RF	RandomForest
RNN	Recurrent Neural Network
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VAV	Variable Air Volume
VPN	Virtual Private Network
XAI	Explanable Artificial Intelligence
XGBoost	Extreme Gradient Boosting

Table of Contents

List of Figures

List of Tables

1. Introduction

The building sector alone is responsible for approximately 36% of the global energy consumption [1]. About half of the energy consumed in commercial buildings comes from heating, ventilation, and air conditioning (HVAC) systems [2], which are used to maintain a certain level of indoor comfort. Meanwhile, common HVAC system faults that are caused by improper maintenance result in 15% of waste in total annual energy consumption [3]. Faults associated with HVAC systems such as sensor faults, control errors, component malfunction, and commissioning flaws can lead to indoor thermal discomfort, reduced component lifespan, and increased energy consumption. Therefore, proper control and maintenance are needed to ensure indoor comfort, increased energy efficiency, and prevent damage of related equipment.

Recently, a growing number of research studies have focused on the development of automated fault detection and diagnosis (FDD) tools for building HVAC systems [4], [5]. The fault detection system is responsible for determining whether the equipment is functioning under normal or faulty conditions, whereas fault diagnosis aims to identify the type or nature of a fault. Another important component is the fault impact evaluation, which involves estimating the severity and consequences of faults to help human operators decide whether or not to take certain actions.

The three common techniques for HVAC system fault detection and diagnosis employed in existing literature can be generalized into knowledge (or rule)-based, model-based, and data-driven methods [4]. Rule-based methods utilize simple if-then rules to identify faults. Such methods may work for simple systems as the rules are defined based on expert knowledge. However, the rules can be difficult to maintain as the number and complexity of faults grow. In addition, it relies on good threshold values to be able to efficiently determine the faulty conditions. The model-based methods require the development of physical models, which can be mathematical representations of real systems. This method allows for simulations and analysis of the fault behavior better than other methods. This can provide engineers with accurate fault diagnostic information. However, developing an accurate physical model of the real system requires extensive expert knowledge, therefore they are time-consuming to develop for complex systems. One problem is that the information needed for the mathematical model is not always available.

Real world systems are difficult to represent by mathematical functions, and the number of components for modeling is huge for large-scale systems. Another method applied in the fault detection domain is the data-driven methods. This method has an advantage over the previous methods as it doesn't require extensive domain knowledge and only uses historical data to process. Modern building management systems allow to collect massive amounts of data, enabling the implementation of more sophisticated data-driven algorithms [4]. Such methods have already become prevalent in industry due to the ability to leverage large amounts of raw data [6] across domains without the need for complex modeling.

Numerous works have implemented a data-driven approach for HVAC fault detection tasks with exceptional results. Some of the work adopted statistical-based approaches, such as PCA and Fisher Discriminant Analysis [7], [8]. Others have tackled the problem with machine learning based approaches, such as convolutional neural network (CNN), artificial neural network and SVM [9]–[13]. The number of publications that focus on artificial intelligence methods has grown more and more in recent years, exceeding those that adopt knowledge-based methods [14].

While data-driven FDD models clearly have ample potential when applied to complex HVAC systems, they may lack the ability to explain and convince users to take action towards energy efficiency. It has the ability to capture non-linear relationships between features and can make accurate predictions [14]. However, it can be the case that the models are trained to maximize their performance and accuracy over the train set. It becomes difficult to tell if the high accuracy is due to over-fitting problems. The black-box nature of the model combined with false-positive results could potentially hinder users from trusting the system. Therefore, improvement of the model accuracy and analysis of the model should work in parallel to increase its reliability.

One way to resolve this issue is to make the machine learning model more transparent [15]. Recent years have witnessed an increasing interest in explainable AI (XAI) research in transportation, healthcare, legal, finance, and engineering domains [6], [16]–[18]. With artificial intelligence dominating in major fields, it becomes imperative to create AI models that are transparent in the sense that the user is presented with an explanation of why the model generated a certain output or made a specific decision, all while preserving high performance and accuracy qualities of the model.

1.1 Background

In recent years, regulators have geared the focus towards the transparency of AI in decision making. The European Union General Data Protection Regulation (GDPR) introduced data protection and privacy act, along with the concept of the rights to explanation [19]. DARPA [15] initiated the Explainable AI program, a set of processes implemented for an AI system to be able to explain itself, thus creating a model that is easily understandable for engineers as well as users and creating trust while maintaining a high learning performance. Explainability adds a layer that helps transform a machine learning model, which is a black-box, into a model that is comprehensible.

This thesis is a joint research project by the research institute of Tallinn University of Technology in conjunction with R8Technologies OÜ. All the dataset used in this work were granted permission by the company for research purposes. The goal of this research is to improve the existing fault detection and diagnosis solutions towards the XAI concept of explainability and transparency.

1.2 Problem Overview

Regardless of the many research studies conducted on fault detection and diagnosis methods for building energy systems, there are still shortcomings and challenges that require attention when developing the diagnosis models, and we will summarize them in this section.

1.2.1 Challenges of FDD for Large-Scale Building Energy Systems

Development of FDD models for HVAC systems may involve some level of uncertainty, which may impede the adoption of the methods despite their high potential. The HVAC system consists of multiple components interacting simultaneously with each other, which makes the system control rather complex. Faults can be distinguished into component faults and sensor faults. Faulty sensors can have an influence on the control loop and lead to undesirable behaviour in the system. In fault detection tasks, these two problems are often tackled separately [14]. Moreover, faults can be caused by events that happen during transient states, which might last only temporarily before the system transitions into steady-state conditions. This may cause false alarms if the fault detection model does not handle the transient-state effectively. Furthermore, the information for fault diagnosis might be incomplete or have some level of uncertainty. Due to sensitivity to sensor costs, some building managers decide to install the minimum number of sensors needed only for control, so information for fault detection may not fully exist. Measurement uncertainty and inaccurate knowledge are also a major challenge in fault diagnosis systems. Additionally, in practice, multiple faults can occur simultaneously. Each system component has its own fault behavior and fault probability. It is also possible that the fault occurring in one component is the direct cause of the fault in another component. Some faults are associated with component degradation and may occur gradually, while some faults occur suddenly and with enough impact to make them easy to notice [4], [5], [7], [14].

1.2.2 Challenges of Data-Driven FDD Methods

Data-driven FDD methods face the challenges of false alarms and a lack of transparency. This method has the advantage of being able to recognize the patterns in the data both automatically and with high performance [6]. However, many works have focused on the performance in terms of accuracy in the data set without the analysis of how well the model represents the actual working of the system. High accuracy can be the result of over-fitting in the train data that is sometimes overlooked after getting a high result score [14]. Another limitation of such black-box models is that they cannot provide meaningful information besides the output prediction. Technicians may need more details in order to make further decisions. Simple regression models are easily interpretable, but they also make less accurate predictions. A lot of state-of-the art machine learning models can make very accurate predictions, but it compromises the interpretability [16].

1.2.3 Explainable Artificial Intelligence in the FDD Pipelines

Although state-of-the-art explainable frameworks [6], [20], [21] exist for improving transparency and gaining trust, their concept is still difficult to grasp by non-technical users. For fault detection problems, explanations are given to end-users on top of the fault predictions to lessen the impact of false alarms. The explainable techniques make it possible to provide explanations for the individual predictions as well as the model as a whole. LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanation) [22], [23] are the most common techniques, which have been applied in the FDD problems for various HVAC systems. However, there remains a gap in the delivery of the explanations that can impede the adoption of XAI methods. State-of-the-art XAI techniques are designed for machine learning engineers across various domains. They are beneficial to machine learning developers in understanding if there's any bias in the training data or if the model learns to capture the correct feature relationships. The XAI outputs can provide insights into the black-box models, but they are still used to communicate with AI experts. Too complicated explanations may cause non-technical users to use the system less. Therefore, it needs to be processed further before useful information can be extracted and delivered to target users. Although data-driven models should be less reliant on experts, their domain knowledge is still necessary throughout the development of the FDD pipeline. Prior knowledge is useful in order to tailor the model interpretation to a meaningful one.

Another challenge in XAI is how to create explanations that are trustworthy and accurate. A misleading explanation can cause users to get the wrong idea or lose confidence in using the system. If the machine learning model is developed using noisy data, the explanations generated from the model may also be of similar usefulness. Therefore, developing a machine learning model and understanding the reasoning behind the model should go in parallel.

1.3 Motivation

The motivation of this thesis stems from the many benefits of having reliable fault detection models for building energy systems. For the building sector, a reliable and transparent fault detection and diagnosis tool will benefit the building management personnel by saving diagnostic time and receiving less complaints from occupants or tenants. The facility manager may benefit from reduced electricity costs and prolonged lifespan of the equipment. Stakeholders involved would also have better insights into the system. Indoor thermal comfort also plays a role in occupants' quality of life and productivity. If the indoor space is well-maintained, occupants can also enjoy the well-conditioned and healthy environment.

Second, an efficient and transparent fault diagnosis model will contribute in achieving environmental and energy sustainability. It will help to move a step towards the concept of near-zero energy buildings (nZEB), which is a measure to enforce environmental and energy sustainability for building management systems. Many countries have already taken the initiative and implemented the concept for new buildings [1].

A lot of tools have been developed to support the energy saving effort. Machine learning models have been utilized in different aspects of building information systems. However, efficient energy saving is not fully possible without the cooperation from end-users. With the rise of AI incorporated in building management systems, the adoption of explainability could play a role in engaging users in the loop of energy saving decision makings [24]. This motivates the development of transparent models for HVAC fault detection and energy efficient solutions.

Lastly, among the papers related to the data-driven approach in building energy systems, only a very small percentage have focused on the explianability aspects. It shows a gap between the technical solutions and the knowledge delivery to end-users. Thus, more attention should be drawn to this area in order to allow building energy tools to become more user-oriented and less of a black-box.

Air handling units will be the focus of this study because they are the main component of the HVAC system. AHU is responsible for regulating the temperature and humidity of the indoor air and ensuring indoor thermal comfort. It is a complex system that functions in various climate conditions. Large commercial buildings usually consist of many AHUs that operate across different floors and zones in the building [5].

1.4 Goal and Research Questions

The goal of this research is to demonstrate the applicability of explainable machine learning methods to aid in the data-driven FDD pipeline for air handling units. The approach could be generalized to other HVAC components as well. We aim to improve the comprehensibility of the predictions made by an XGBoost model by applying an interpreter method called SHapley Additive exPlanation (SHAP). XGBoost is a machine learning method that has the ability to capture nonlinear relationships in tabular data and it has been applied in the FDD domain [6]. Therefore, we would like to explore the potential of XGBoost, by comparing its accuracy to the classic regression model and a baseline RandomForest model. With the aim of involving end-users in the decision-making, we provide the explanations for the AHU data that was observed from a real commercial building. In this study, fault detection and diagnosis will be implemented in a single step, where the fault diagnosis model will diagnose the fault classes as well as the normal class.

To accomplish the above goal, we seek to answer the following research questions:

1. Fault detection and diagnosis
   (a) Using data-driven methods, how can we detect and classify common faults in the air handling unit?
2. Model output explanations
   (a) What methods can we use to explain the model output prediction of the fault samples?
   (b) How can we communicate effectively to the target users (in this case, the HVAC engineers) the model predictions of the fault instances in a way that they can easily comprehend?
   (c) How can we include extra information or context that helps in understanding the nature of the fault that occurs?
   (d) How can we evaluate the provided explanations?
   (e) Are there other ways to improve users' trust in the system?

1.5 Contributions

This paper presents a method to explain the decision of an XGBoost-based classifier using SHAP as the interpreter. The method could benefit the end-users requiring justification and reasoning behind the fault predictions. For interpretability in the context of HVAC fault detection with time-series data, we visualized sliding-windows to provide explanations for the observed faults. The aim is to provide insights into not only what happens in a single time step, but also to understand what has happened prior to the observation that leads to the fault occurring. To keep the explanation information focused, we only showed the user-defined sets of features corresponding to the observed fault and the features with high contribution scores. The experiment was conducted on real-world data of an air handling unit containing normal and faulty samples. The explanations were applied to samples of each fault type and then assessed by the HVAC experts.

To summarize, the main contributions of this work to the research area of explainable fault detection and diagnosis methods for building energy systems are outlined as follows:

1. We developed a method to explain the fault diagnosis output of an XGBoost-based model using Shapley values. We use sliding-windows to visualize the short history of the relevant features and to provide explanations for the diagnosed faults in the observed time period. This provides the ability to understand not only what happens in each individual time step but also to monitor the progress history of the fault. The fault detection and diagnosis pipeline is conducted using real-world data obtained from an air handling unit of a commercial building.
2. We propose a method to incorporate human users into the decision making by allowing the selection of relevant features to be explained for each fault type. The method explains the features with high Shapley values and features corresponding to each fault type. This could provide practical value by keeping the explained features relevant.
3. We present the analysis for XAI explanations of each fault type in the AHU dataset, by using domain expert evaluation to obtain feedback on the generated explanations. This helps us understand the effects of the explanations on the users' decision-making and how well users can understand the explanations.

1.6 Structure of the Thesis

This thesis is organized as follows: Chapter 1 introduces the importance of fault detection and diagnosis for building energy systems, identifies the challenges, describes the motivation behind the thesis, and elaborates on the contributions. Chapter 2 includes a literature review on existing fault diagnosis methods and XAI techniques used in building applications. Chapter 3 provides technical background on air handling units and describes some of the data-driven techniques applied in the literature as well as XAI techniques that have been applied in HVAC fault detection. Chapter 4 describes the research methodology, evaluation metrics, and justifications for using the selected XAI method. Chapter 5 provides descriptions of the data sets used in the study and the implementation of the model. Chapter 6 reports the numeric results and analysis, provides the interpretation for each example fault case, and focuses on the expert survey. Discussions are drawn in Chapter 7. Finally, the work is concluded, and directions for future work are presented in Chapter 8.

2. Literature Review

This section covers literature reviews on FDD and explanable AI methods. The discussion is divided into two parts. The first covers the fault detection and diagnosis methods for HVAC systems, with more emphasis on the data-driven methods. The second part examines the explanable AI studies that have been applied to the building applications. This includes XAI applied to general building applications as well as the FDD domain.

2.1 Fault Detection and Diagnosis for HVAC Systems

Fault detection and diagnosis has drawn attention as many methods have been proposed from different approaches to solve this task. Extensive literature reviews of building system fault detection and diagnosis are conducted in [4] and in [5]. In this section, we will discuss the three main strategies, which are rule-based, model-based, and data-driven methods, of which the data-driven method will be the main focus.

Rule-based methods utilize knowledge of if-then rules and expert knowledge of control systems to classify faulty and non-faulty states. AHU Performance Assessment Rules (APAR), proposed in [25], determine AHU operation modes based on control signals, setpoint values, sensor measurements, and occupancy information. Control signals are used to determine the mode of operation, under which 28 rules are specified. If a rule evaluates to true, a fault is implied. In [26], a hierarchical rule-based method has been applied to AHU. The method aims to suppress and reduce the number of false alarms by detecting source faults when multiple load faults occur simultaneously. This enables detection of faults otherwise ignored by models that only look at individual components. However, the problem with rule-based methods is that the number of if-then rules may grow so large that it becomes very hard to maintain for large systems. Rule-based methods may become a disadvantage when the information or the sensor measurement is not available. When a sensor crucial to the fault rules is missing, further if-then rules need to be written to handle such scenarios. Therefore, the complexity of the rules can grow very fast.

Model-based methods rely on precise modeling of the plants into a physical model. This type of method uses the difference between the real measurement value and the value obtained from the modeling process. The difference is regarded as the residual, which works as an indication of fault. Faults may occur if the residuals cross pre-determined threshold values. Accurate modeling of the physical system can give precise diagnostics of system faults. However, as mentioned in earlier sections, this technique is very time-consuming to develop, especially for large-scale systems [4]. Extensive knowledge of control systems is required in order to model the entire system into a physical model. The required information is not always available, or the system can be too complex to model accurately. Therefore, this method is better suited for simpler systems [4]. With model-based methods, solving for good threshold values is crucial to determining faulty conditions. Threshold values too high will cause the model to miss fault alarms, while threshold values too low will make the model very sensitive to slight fault conditions and can generate too many false alarms.

Data-driven methods use historical data to process the faults. Therefore, this type of method relies less on prior knowledge. The method has gained popularity in HVAC system fault detection and diagnosis in recent years. The authors of [7] and [8] have adopted the principle component analysis (PCA) method to detect faults in air handling units. PCA methods have been widely applied in detecting sensor faults [14]. It is an unsupervised learning approach used in fault detection tasks. Although the method is reported to have promising results, it does not retain the original feature relationships, limiting its practicality for fault diagnosis tasks where important features need to be identified in order to locate the root causes of the faults.

Some studies focus on machine learning methods such as support vector machines (SVM) [12], [27] and neural networks (NNs) [10], [11], [13] for solving FDD tasks. While many studies focus on the diagnosis of faults that appear one at a time, the authors of [12] use SVM combined with the multi-label classification approach to tackle the problem of multiple simultaneous faults in building chillers. The faults are related to reduced condenser water flow and reduced evaporator water flow. The model was trained using samples containing individual faults and was able to detect when the combination of the individual faults exists. In [10], a back-propagation neural network is used for fault detection. Fault-free data is used as the input to the network, which is trained to predict the value of the control parameter. Based on the prediction, the residual is calculated and is compared with a predefined threshold value to determine the fault presence.

Deep learning methods such as convolutional neural networks (CNN) have received increasing interest for FDD problems due to their high performance, computational efficiency, and ability to perform feature extraction and classification simultaneously [9], [28]. In [9], a fault diagnosis model was proposed using a one-dimensional CNN (1D-CNN) model. Statistical methods were used to diagnose sensor faults, and 1D-CNN was used to diagnose four AHU faults in cooling coil valves, fan circuits, ducts, and outdoor air dampers. The applicability of 1D-CNN is also demonstrated in [20] to diagnose seven chiller faults.

2.2 Overview of XAI Techniques in Building Applications

In this section, the application of explainable AI techniques to general problems in buildings is first discussed. The focus is then shifted to the specific issues related to fault detection and diagnosis in typical technical units. Based on the literature review, we argue that the use of XAI for building applications is still new, and only a few studies have been reported so far.

The general applications mostly encompass common problems of evaluating building performance and predicting energy demand. In [29] XAI techniques were applied to the XGBoost model for long-term forecasting of the cooling energy consumption of buildings located in different climatic areas. Another explainable long-term prediction model was introduced in [30], which studied predictions of annual building energy performance. In [31]–[33], the authors focused on developing attention mechanisms to improve the interpretability of the developed models. The benchmark of buildings using explainable AI was addressed in several recent papers [34]–[36]. In [37], the use of explainability techniques was proposed in the context of smart home applications, while [38] focused on a more general smart city concept.

Several recent papers on fault detection and diagnosis for HVAC systems have focused on explainability for gaining user trust. XAI methods have been adopted to visualize model output predictions of individual samples. Fault samples are classified using machine learning models and XAI methods are used to explain individual samples by visualizing the contribution of each feature to the final output. This can help users understand whether the prediction should be trusted. In [6], the LIME (Local Interpretable Model-agnostic Explanations) framework was adopted to explain cases of incipient faults, sensor faults, and false positive results of the diagnosis model for the chiller system, which is based on the XGBoost model. The general XAI-FDD workflow was validated using several real test cases. The proposed approach allowed to reduce manual fault-detection time, analyze the sources and origins of the problems, and improve maintenance planning. The authors of [21] used the LIME method to explain the fault classification results of the support vector machine and neural network models developed for the diagnosis of heat recycler systems. The method was able to explain the diagnosis of the component faults using examples of individual instances. In [20], a new Absolute Gradient-weighted Class Activation Mapping (Grad-Absolute-CAM) method was proposed to visualize the fault diagnosis criteria and provide the fault-discriminative information for explainability of the 1D-CNN model, applied to the detection of faults in chiller systems. The developed method was validated using an experimental dataset of an HVAC system, showing high diagnosis accuracy for seven chiller faults. The proposed method was able to successfully explain all the fault criteria.

The above mentioned approaches have demonstrated significant improvement from the traditional data-driven fault diagnosis pipelines where the end result is the fault class. In the new approach, an additional layer is implemented, which is the interpretation of the sample prediction. Therefore, the data-driven models are no longer black-box but are transparent and comprehensible by end users.

Table 1 provides a summary of the XAI concept used in buildings applications. These can be divided into two classes: methods that solve fault detection and diagnosis problem in specific units, and methods that target general building applications. All these works report a significant improvement in understanding results obtained using machine learning models.

Table 1. Summary of explainable AI methods in buildings applications.

	Ref.	Application	AI Model	XAI Technique	Year
FDD	This Work	Detecting AHU faults	LogisticRegression, RF, XGBoost	SHAP	2022
	[6]	Detecting incipient, sensor, and chiller faults	XGBoost	LIME	2021
	[20]	Detecting chiller faults	1D-CNN	Grad-Absolute-CAM	2021
	[21]	Detecting heat recycler faults	SVM and NN	LIME	2019
General Applications	[30]	Predicting long-term building energy performance	QLattice	Permutation feature importance	2022
	[29]	Analysis and prediction of climate change impacts on building cooling energy consumption	XGBoost	SHAP	2021
	[39]	Performance forecast of irregular dew point cooler	Deep Neural Network	SHAP	2021
	[33]	Short-term forecasts of building energy consumption	Encoder Decoder model with RNN sequence	Attention mechanism	2021
	[35]	Classification of building energy performance certificate rating levels	ANN	LIME	2021
	[34]	Benchmarking building energy performance levels	XGBoost	SHAP	2020
	[36]	Identifying usage patterns and building energy performance	Classifier using ML	Correlation among temporal features	2019
	[40]	Predicting coefficient of performance of the cooling system	SVM, MLP, XGBoost, RF	LIME	2019

3. Technical Backgroud

In this section, we provide technical information about the air handling unit system and the fault diagnosis methods used in the analysis. We will focus on data-driven fault diagnosis techniques and explainable AI techniques.

First, let us review the basics of the workings of an air handling unit.

3.1 Air Handling Units

AHUs are commonly used in commercial and residential buildings in order to maintain indoor conditioning. Figure 1 shows a schematic diagram of a variable-air-volume (VAV) air handling unit utilized in this study. The AHU consists of fans, dampers, cooling and heating coils, sensors, controllers, and heat recovery units. The supply fan draws fresh air from outside, and the air is passed through a fan filter to filter dust or objects that could cause damage or inefficiency inside the system. Then, the heating coil and cooling coil valves modulate to be open or closed in order to maintain the supply air at the desired setpoint temperature. For example, when the supply air temperature is too low, the heating coil valve opens to allow hot water to flow through the heating coil and heat up the supply air until the setpoint is reached. The conditioned supply air is then distributed to different zones or rooms in the building. Several factors can affect indoor air quality. If the room is occupied, heat might radiate from the body and cause a high room temperature. It may also affect the CO_2 and humidity levels, and the room may become uncomfortable again. Therefore, the air in the room needs to be circulated back while letting fresh air in. The return fan draws the air from the zone back to the unit. Then the return air is either recirculated and mixed with the fresh air in the mixing box or is drawn away as the exhaust air [5], [8], [10].

In order to maintain energy efficiency, the heat recovery system is utilized to recycle the return air by mixing it with the fresh intake air in the mixing box. That way, some amount of energy can be recovered without having to activate the heating system every time heating is required, since the heating system is more energy demanding than using heat recovery. In the ventilation unit used for this study, the heat recovery system is a rotary system. The efficiency of the heat recovery is an indicator of the amount of energy that it manages to recycle [5], [21].

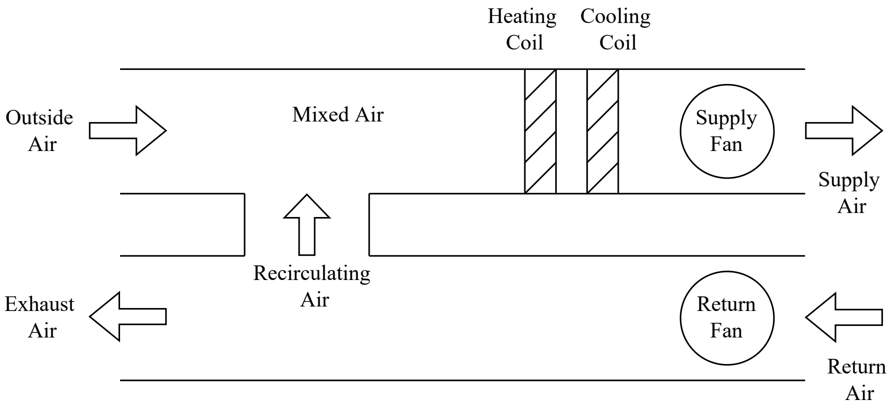


Figure 1. The schematic diagram of an air handling unit.

3.2 State-of-the-art Data-driven Techniques

A lot of research has focused on faults in individual HVAC components, such as chillers, heat recovery units, air handling units, heat pumps, and sensors [6], [7], [10], [12], [28]. A comprehensive review on HVAC data-driven fault detection and diagnosis methods is presented in [14] and [4]. Data-driven methods have dominated the research field, constituting 79% of the total research papers published in the fault detection and diagnosis of building energy systems. We will describe the methods used in this study as well as the alternative methods that were mentioned in the literature.

3.2.1 Logistic Regression

Logistic regression is a statistical model based on regression analysis and is used for binary classification problems. It attempts to find the best-fitting model that describes the relationship between the dependent variable, i.e., the feature of interest, and independent variables. It maps the output of a linear model into a value between 0 and 1, using the logistic function [41]. The logistic regression can also output the probability P [41], which can be useful for fault detection, because further action can be taken based on the fault probability. The logistic function, or the sigmoid function, is defined as:

$$f(\eta) = \frac{1}{1 + e^{-\eta}}. \tag{3.1}$$

For classification, we can get probabilities between 0 and 1 by wrapping the linear model into the logistic function. A logistic regression model can be written as:

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}}, \tag{3.2}$$

where $P(Y = 1)$ is the probability of the observed class, β_0 is the intercept term, $x_1, x_2, \ldots, x_p$ are the inputs for each feature, $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficients of the linear model corresponding to each feature input. The input vectors can also contain both continuous and categorical variables [41].

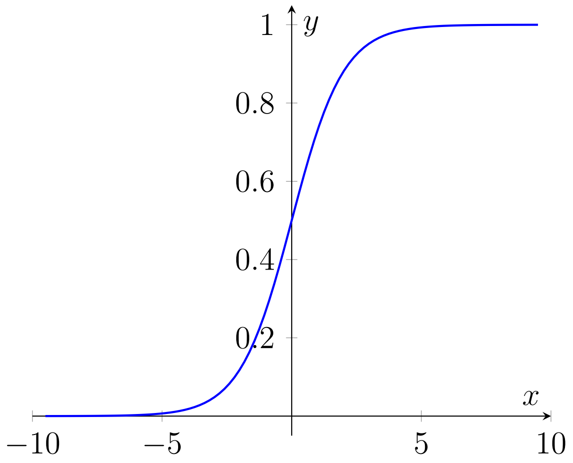


Figure 2. Standard logistic function.

3.2.2 Random Forest

RF is an ensemble machine learning method which works by averaging multiple decision trees. The tree of RF is trained through random variable selection to create clusters of decision trees during training time. In the development of classic classification and regression trees, the selection of split variables is strongly influenced by the distribution of samples in the train set, which may cause over-fitting. The goal of RF is to reduce the over-fitting problem that can be caused by using only a single decision tree on the training set. RF overcomes this by averaging multiple trees to obtain the output. In classification tasks, RF outputs the result voted by most individual trees. In regression tasks, it outputs the average predictions from each individual tree [42]. RF is a black-box method as the depth and the number of the trees can become complex through many possible configurations.

Figure 3 illustrates the modeling process of RF. First, the process starts by generating new subsets of training samples through replacement, which is the same as the bagging approach. On average, two-thirds of the original samples are used to train the tree, while the rest are for internal cross-validation. Then the features are sub-sampled randomly to create a new set of features K from the original feature set. Instead of using the whole feature sets, the new subsets are used to train the tree. Then each individual tree in RF uses the new randomly generated feature set and the new training subsets to find the best split and grow the tree. Each tree is developed independently during training time. After all the trees are developed, RF gets the final result by averaging the output or taking the majority votes from all the trees [42].

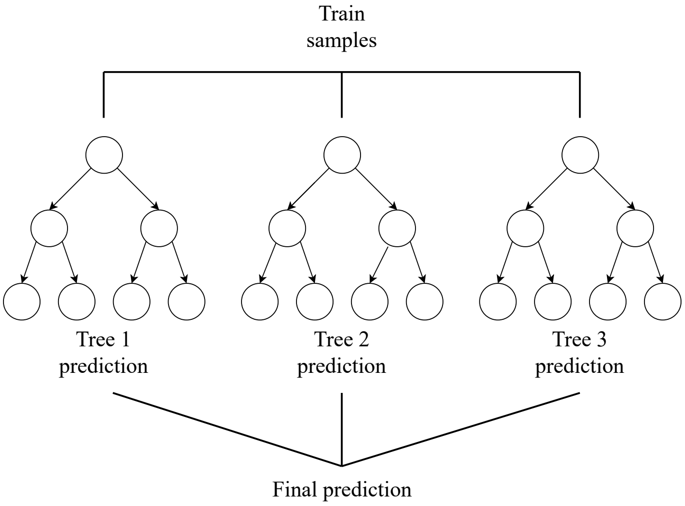


Figure 3. Modeling process of random forest.

3.2.3 eXtreme Gradient Boosting

XGBoost [43] is an ensemble model based on Gradient Tree Boosting that works by integrating several basic classifiers together, which are usually decision tree models, to form a more robust model. The model learns through an additive manner or a cumulative learning process. First, the starting tree is fitted with the entire training data. Then the learning result of the tree is passed to the next tree to update the weights and the process is repeated. The final result is obtained by accumulating the results from all the trees. The prediction function in step t is denoted as:

$$f_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = f_i^{(t-1)} + f_i(x_i), \tag{3.3}$$

where $f_i(x_i)$ is the tree model at step t, $f_i^{(t)}$ and $f_i^{(t-1)}$ are the predicted values in step t and $t-1$. To learn the sets of functions, XGBoost seeks to minimize the following objective:

$$L(\phi) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{m} \Omega(f_k), \tag{3.4}$$

where

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2. \tag{3.5}$$

Here, n is the number of training samples, $l(\hat{y}_i, y_i)$ is the loss function. $\Omega(f_k)$ is the regularization term on the k^{th} decision tree, w is the score from leaf nodes, λ is the hyper parameter of regularization term. γ is the minimum loss that the leaf node needs to make further splits. The regularization term Ω penalizes the complexity of the model and helps smooth the final learnt weight to avoid over-fitting. Without the regularization term, the objective function falls back to the traditional gradient tree boosting [43].

For a faster calculation, XGBoost uses the second-order Taylor's expression, as denoted in (3.6), in the loss function when calculating the objective function.

$$L^{(t)} \simeq \sum_{i=1}^{n} [\, l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] \ + \Omega(f_t) \tag{3.6}$$

3.2.4 Alternative Approaches

Principle Component Analysis

PCA [44] has been widely applied in the fault detection of HVAC systems. PCA is a dimension reduction technique that can extract useful features from high-dimensional data. It works by projecting data points into a lower dimensional space using only the first few principle components (PCs). The goal is to obtain a lower dimensional data while preserving the data variance as much as possible. The first step is to standardize the raw data to have zero mean and unit variance:

$$x_n = \frac{x_n - \overline{X}}{\sigma_n}. \tag{3.7}$$

The next step is to compute the covariance matrix:

$$S = \frac{1}{N} \sum_{n=1}^{N} (x_n)(x_n)^{\mathsf{T}}. \tag{3.8}$$

Then the eigenvectors (PCs) and eigenvalues can be computed based on the covariance matrix. More details on the PCA can be found in [44]. The eigenvectors represent the vector directions of the new obtained features, while the eigenvalues are the magnitudes that correspond to the directions. The eigenvectors are then sorted based on the corresponding eigenvalues, and the first few PCs are selected to obtain a new reduced subspace A.

Support Vector Machine

In SVM [45], the data points are projected into the dimensional space. The goal of SVM in classification tasks is to search for an optimal hyperplane in the sample space where the distance between the hyperplane and the nearest data points of the different classes is maximized. SVM has shown potential in solving practical tasks that involve nonlinearity, and a small number of features and samples.

In its most basic form, the SVM classifier handles data with classes that are linearly separable. It solves for the optimal linear vectors that can separate the two classes. Linear SVM seeks to solve the function $wx + b = 0$ to find the hyperplane that maximize the separation distance [45].

One-Dimensional Convolutional Neural Network

CNN is a type of deep neural network widely applied for pattern recognition, namely image and speech recognition. Architecturally, CNN is a feedforward neural network that consists of the input layer, convolutional layer, pooling layer, fully connected layer, and output layer. After the features are passed as the input, the convolutional layer, which is the core of CNN, performs feature extraction. Then, the extracted features are flattened and passed to the fully connected layer, which works similarly to a neural network. Finally, the activation function is applied and the classification result is obtained [9].

In HVAC fault detection, 1-D CNN has been proposed and successfully applied to detect faults in AHUs and chillers [9][20]. The goal is to sequentially extract the relevant features from the original feature sets of the time-series HVAC data. The input is one dimensional, so the convolution kernel and feature map are also one dimensional [9].

3.3 XAI Techniques

For a very simple model, it is possible to use the model itself as the explanation [23]. A linear regression model can be explained using the feature weights. For tree-based models, the relationship between features can be visualized in terms of a tree structure, where the root node branches out into branches and leaves. However, with different configurations, the depth of the tree can grow very large and the tree can become very complex, which causes the tree structure to become incomprehensible. This problem creates a trade-off between model prediction accuracy and model interpretability. Complex machine learning models such as support vector machines, artificial neural networks, etc. can make predictions with very high accuracy but are black-box in nature [16]. Nonetheless, it is crucial to understand the rationale behind the decision making process taking place in the machine in order to invite more human involvement into the loop and obtain more trust along the way. Many methods have been developed for explaining machine learning models, such as LIME (Local interpretable model-agnostic explanations) [22], SHAP (SHapley Additive exPlanation) [23], CIU (Contextual Importance and Utility) [46], and GradCAM (Gradient-weighted Class Activation Mapping) [47], in which the input can be an image, text, tabular data, etc. [48]. Among the above mentioned methods, LIME and SHAP have emerged as the most common techniques that have already been around for a number of years.

Figure 4 depicts the schematic flow of a general process dedicated to the generation of explanations for AI-based models. Here, an additional "Explainer" layer is used at the later stage to generate explanations by highlighting the main features that are significant for the model output and to present them in a form comprehensible by the end user.

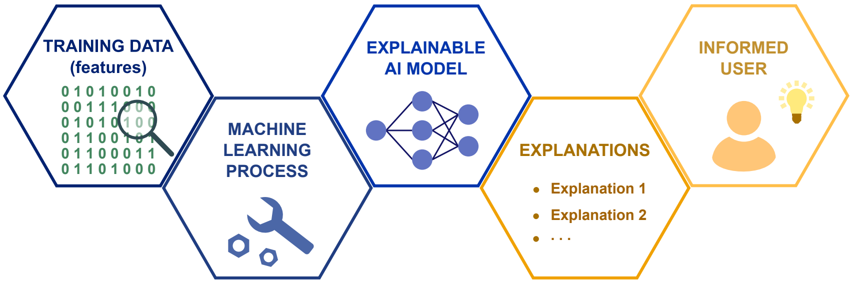


Figure 4. The schematic of a conceptual XAI framework with an additional explanation module, aiming to bridge the gap between decisions made by a model and user.

Two ways to comprehend a black-box machine learning model are through local and global interpretations. Global interpretations allow understanding of the entire model, while local interpretations enable understanding of individual predictions. Local interpretation is used to justify why a model generates an output for a specific instance [16]. For fault detection tasks, local explanations are generated to justify why the model classifies an individual sample as faulty [6], [20], [21].

XAI techniques can be further classified into model-agnostic and model-specific. Model-agnostic methods are methods that can be applied regardless of the machine learning model used. In contrast, model-specific methods are tied to specific models. It can even be tied to a specific structure of a neural network. Neural networks may contain many layers and a large number of weights that have complex interactions. Specific methods might be required to explain the gradients of the neural network layers [16], [41].

3.3.1 SHapley Additive exPlanation

SHAP [23] is a game theory-based approach to explain the individual predictions produced by machine learning models. It is used to show the contributions of the input features using the computed Shapley values, where each feature works together as a coalition. The Shapley value is calculated for each feature in the input samples that needs to be explained. Based on the aggregated Shapley values, it can also provide global interpretations of the black-box models. SHAP describes three desirable properties, which are local accuracy, missingness, and consistency [23].

1. **Local Accuracy**
   When approximating the original model f for an input x, local accuracy is the ability of the explainer model to represent the output of the simplified model f' for the simplified input x'.
   $$f(x) = g(x)' = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{3.9}$$
2. **Missingness**
   Missingness requires that the features that are missing from the input to have zero impact on the model output.
   $$x'_i = 0 \rightarrow \phi = 0 \tag{3.10}$$
3. **Consistency**
   Let $f_x(z') = f(h_x(z'))$ and $z'\backslash i$ denote setting $z'_i = 0$. For any two models f and f' if
   $$f'_x(z') - f'_x(z'\backslash i) \geq f_x(z') - f_x(z'\backslash i) \tag{3.11}$$
   for all inputs $z' \in 0, 1^M$, then $\phi_i(f', x) \geq \phi_i(f, x)$. The consistency property states that if a model changes so that the marginal contribution of a feature increases or stays the same regardless of the other inputs, the attribution of that input feature should also increase or stay the same.

The formula for calculating the Shapley value for a feature is denoted by:

$$\phi_i = \sum_{S \subseteq F\{i\}} \frac{\mid S\mid !(F - \mid S\mid \; - 1)!}{\mid F\mid} [f_{S\cup\{i\}} - f_S)], \tag{3.12}$$

where S represents the feature subsets, and N is the set of all features. To compute the effect of a feature, a model $f_{S\cup\{i\}}$ is trained with that feature present. Another model f_S is trained without that feature. Then, the outputs from the two models are used to calculate the difference. The difference is calculated for each possible subset $S \cup F$, then we obtain the attribution of that feature, which is the weighted average of all the calculated differences.

The author of SHAP proposed different variations of SHAP. Kernel SHAP [23] estimates the feature attributions for individual instances using weighted linear regression. The problem with Kernel SHAP is that it requires the computation of Shapley values for all the features. The computation time grows as the number of features and size of the dataset grow. Therefore, it is slow to compute and impractical for real-world tasks [23], [41]. An alternative to Kernel SHAP is Tree SHAP. Tree SHAP uses conditional expectations to estimate the feature effects. Tree SHAP is a fast and model-specific method to calculate SHAP values from tree models such as decision trees, random forests, and other ensemble tree models. The computational complexity for Tree SHAP is $O(TLD^2)$. As a comparison, Kernel SHAP complexity is exponential: $O(TL2_M)$, where T is the number of trees, D is maximum depth of any tree, L is the maximum number of leaves in any tree, M is the number of explained features [41].

3.3.2 Alternative Approaches

Local Interpretable Model-Agnostic Explanations

LIME [22] is a method for interpreting individual predictions made by machine learning models. It works by locally approximating the model around a given prediction. LIME seeks to minimize the following objective function:

$$\xi(x) = \underset{g \in G}{\arg\min}(L(f, g, \pi_x) + \Omega(g)), \tag{3.13}$$

where f is the original model that needs to be explained, ξ is the local explanation for sample x, g is any model from a class of interpretable models, which can be linear regression or decision tree model, $\Omega(g)$ is a measure of the complexity of model g, and $\pi_x(z)$ represents the proximity measure between an instance z to x. The goal is to minimize the loss function L which measures how accurate the explanation reflects the prediction of the original model. To achieve both interpretability and local fidelity, LIME minimizes the loss while keeping the complexity low at the same time [22].

Gradient-weighted Class Activation Mapping (Grad-CAM)

Grad-CAM [47] is a model-specific explanable method that works with CNN models. The method uses gradient information at the last convolution layer of the target class to enable visualization of important feature map activations. To obtain the feature map, the first step in Grad-CAM is to compute the gradient with respect to the feature map activation A of a convolutional layer. Then, the second step is to obtain the neuron importance weights by averaging the computed gradients [47]:

$$a_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\delta y^c}{A_{ij}^k}, \tag{3.14}$$

where δy^c is the gradient of score for class c and the weight a_k^c represents a partial linearization of the deep network downstream from A, and captures the 'importance' of feature map k for a target class c. The last step is to pass the weighted activation maps into ReLU function to compute the final Grad-CAM heatmap [47]:

$$L_{Grad-CAM}^c = ReLU(\sum_k a_k^c A^k). \tag{3.15}$$

In fault detection and diagnosis tasks, the Grad-CAM concept has been adopted for one-dimensional CNN models for tabular data. Another variation of Grad-CAM has been proposed [28].

4. Methodology

This section outlines the methodology for developing the XAI-based FDD pipeline as well as assessing the explanations of the black-box fault diagnosis model. The major steps in the process include: data collection and pre-processing, XGBoost-based fault diagnosis model, SHAP explanations, and expert evaluation of the explanations.

This research study is organized as follows:

1. A fault detection and diagnosis model based on the XGBoost classifier is implemented and compared with two baseline models, which are logistic regression and random forest models. The FDD process is done in a single step, with the diagnosis model classifying the five types of faults from the normal operation state.
2. A case study was conducted using real data collected from a commercial building (a shopping mall) located in Estonia. The data pre-processing and feature engineering steps are described along with the description of what each fault type means and what are the features corresponding to each fault type based on prior-knowledge.
3. Each model result is summarized using various evaluation metrics, including the F1 score, to compare the accuracy results on the test set. The score is used as a criteria to decide on which model to use as our fault diagnosis model.
4. The chosen models will be analyzed based on the global interpretation, using the SHAP summary plot.
5. Samples of five different types of faults in the air handling units are selected to provide explanations of the model. The SHAP method is integrated as the explanation algorithm. The output of SHAP is visualized and analyzed for each fault case to understand the reasoning behind our model predictions for each individual fault sample.
6. Different visualizations are provided where the outputs of SHAP are visualized using sliding-window observations instead of single time step observations. This will give a short history of the measurement overtime until the fault occurs or disappears. Only the features corresponding to each fault type and features with high SHAP values are visualized. The mapping of feature sets for each fault type will help to keep the number of features relevant to communicate more effectively to end-users.
7. The generated explanations are then assessed by certified HVAC engineers possess-

ing expert knowledge, who will give feedback and help in evaluating the effectiveness of the explanations on their decision making.

Figure 5 outlines the proposed methodology, which can be summarized as follows:

1. Offline model training stage:
   (a) Data is collected for faulty and fault-free operations and is labeled according to the fault types. Samples that don't belong to any fault class are labeled as normal. Data is pre-processed by removing records with null or non-existing values. Samples during off-state and during the first hour of operation are also removed.
   (b) Prior knowledge related to all fault types is gathered. This includes the mapping of feature sets that are corresponding to each type of fault. The feature sets can be inputted by end-users, who decides which relevant features they want to see for each type of fault, or else the default feature sets will be chosen.
   (c) An XGBoostClassifier model is implemented for the FDD problem. The model is a multi-class multi-label classification model, which is used to classify which fault class(es) each sample belongs to. One sample can belong to multiple fault classes.
   (d) SHAP method is used to generate explanations for the fault diagnosis model. A Tree SHAP explainer is fit using the developed model to be able to generate explanations during online monitoring stage.
2. Online fault monitoring stage:
   (a) Real-time measurement is obtained from the system. The new observation is pre-processed and input into the trained XGBoost model to classify between the fault classes and the normal class.
   (b) If the sample represents a faulty operation, the interpreter module is triggered to generate the explanations. If no fault is detected, skip the following steps.
   (c) Using the fitted SHAP explainer object from the previous offline training stage, SHAP values are generated for the observed faulty samples and the samples a number of time steps prior to the observation, to provide a short history of the fault occurrence.
   (d) Create visualization for relevant features using a sliding window graph.
   (e) User gives feedback on the explained feature choices. It will be used to update the sets of relevant features.

Multi-Label Approach

In contrast to the mono-label fault diagnosis problems where one sample can only be assigned one class exclusively, our approach is a multi-label approach where one sample

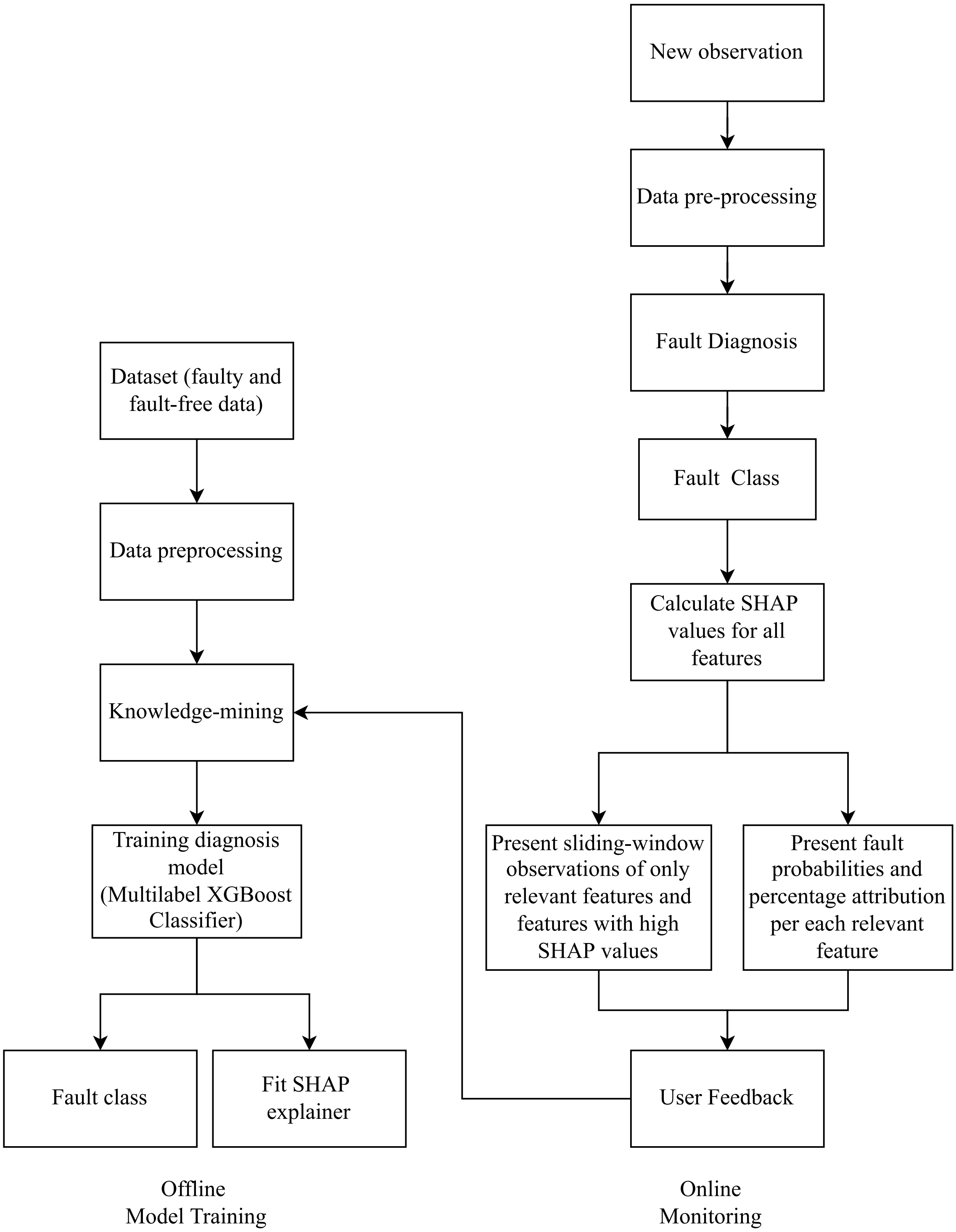

New observation
Data pre-processing
Fault Diagnosis
Fault Class
Calculate SHAP values for all features
Present sliding-window observations of only relevant features and features with high SHAP values
Present fault probabilities and percentage attribution per each relevant feature
User Feedback
Dataset (faulty and fault-free data)
Data preprocessing
Knowledge-mining
Training diagnosis model (Multilabel XGBoost Classifier)
Fault class
Fit SHAP explainer
Offline Model Training
Online Monitoring

Figure 5. Proposed explainable fault detection and diagnosis pipeline.

can belong to more than one fault class at the same time. This allows for the diagnosis of multiple simultaneous faults. Our reason for using multi-label classification is to make the model scalable and to suit practical fault diagnosis tasks. The authors of [12] proposed a fault diagnosis solution using a multi-label classification approach where the model is trained using only samples with individual faults and the model is able to predict more than one fault class. In our work, one classifier is developed and fitted per target class. That is to extend the binary classification to a multi-target approach.

Justifications for Adoption of SHAP in the Explanability Layer

SHAP has a strong theoretical root. SHAP computes Shapley values which is based on a solid theoretical foundation in game theory. So all the advantages of Shapley values apply to SHAP as well. Similar to Shapley value, which describes the outcome by "fairly" distributing the "payout" among all the players, SHAP describes the prediction output by “fairly” distributing the prediction value among the features, based on how much each feature contributes to the final prediction [41].

SHAP has a fast-implementation for tree-based models, TreeSHAP. Since we’re implementing XGBoost, which is an ensemble tree model, it can work well with TreeSHAP. This could provide an advantage in terms of computation speed and resources, and the explanations can be generated in real-time. In addition, TreeSHAP doesn’t require access to the data for calculating the Shapley value for the new observations [41].

With SHAP, the local explanations are consistent with the global explanations since the Shapley values constitute the global explanations. SHAP provides global interpretation methods such as feature importance, interaction plots, and summary plots, which are powerful visualization techniques in understanding the model as a whole [41].

SHAP has been applied in various fields such as medicine, energy systems, and fault detection domain [39], [49], [50]. It can be conveniently applied to any black-box machine learning model and to different types of data, i.e., tabular, image, or text data, since it’s model-agnostic.

4.1 Performance Metrics for FDD

We use the F-measure, precision, recall, sensitivity, specificity, accuracy, and confusion matrix to assess the performance of the classification models. The F-measure (or balanced

F_1 score) is the harmonic mean of the *precision* and *recall* measures, defined as [51]:

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{4.1}$$

where

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4.2}$$

TP is the number of true positives, FP is the number of false positives and FN is false negative.

Precision is the proportion of samples that were diagnosed as faulty samples that actually were faulty samples. In this case, precision is the ratio of the number of samples correctly predicted as a fault to the total number of samples predicted as a fault. On the other hand, the recall is the measure of true fault prediction correctly labeled in the true fault class. In this case, recall is the ratio of the number of correct true fault predictions to the total number of true fault samples [51].

The accuracy is the ratio of the number of correct fault predictions to the total number of predicted samples. Accuracy is denoted as:

$$\text{accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}. \tag{4.3}$$

Sensitivity, which is identical to recall, refers to the proportion of faulty samples that are actually faulty. Specificity is the proportion of fault-free samples that are actually fault-free, were predicted by the model as fault-free [51]. Specificity is defined as:

$$\text{specificity} = \frac{\text{TN}}{\text{TN+FP}}. \tag{4.4}$$

The metrics are shown for each classification model for each fault class, and they are also aggregated into weighted averages to show the overall performance of the models in predicting all the classes. The weighted average accounts for the contribution of each class, which is given different weights depending on the number of samples in that given class.

4.2 Explaining the Fault Predictions

Our challenge is to design the explanations and deliver relevant information that helps target users to identify faults. The target users in our case are the HVAC engineers, as opposed to the machine learning engineers. We implemented a fault diagnosis model to classify the faults in the AHU using a supervised learning approach. If a fault exists, SHAP explainer is triggered to provide the explanations using sliding window method. Here we use Tree SHAP to generate the SHAP values, which describes the attribution score for each feature in predicting the fault classes.

Given a data instance X with features $x_1, x_2, \ldots, x_n$, and a classifier model f, Tree SHAP receives f as the input to get the explainer object exp. Then, exp takes X as the input and generate the SHAP values. The process is depicted in Algorithm 1.

The process to select features for explanations is shown in Figure 5 and described in Algorithm 2. The authors of [49] described the process for selecting features used to explain an autoencoder-based anomaly detection model. In our work, the feature selection for explanations are based on:

1. Features that have SHAP values higher than mean SHAP values.
2. Features that are pre-selected by users or features that are mapped corresponding to each fault type using prior knowledge.

The SHAP values are calculated for each of the feature in the faulty samples. Since raw SHAP values are not easily comprehensible by non-technical users, the values are rescaled into percentage contribution using logistic transformation.

Algorithm 1 Calculating SHAP values

```
Inputs:
X: instance for which the explanations is generated
f: classification model for fault diagnosis
Output:
shapVal: SHAP values for all features
Begin
    explainer ← shap.TreeExplainer(f)
    shapValue ← explainer.shapvalues(X)
    Return: shapValue
End
```

4.3 Expert Interviews on the Explanations

To validate the explanations for the fault diagnosis model, we will conduct an interview with HVAC experts. We chose random fault samples from each fault type in the dataset and generated the explanations using three different visualization techniques: explanations for individual instances based on SHAP, explanations using SHAP force plot, and SHAP-based explanations using the proposed method in this study. Our aim is to assess how well the user understands the model prediction of fault and to explore and compare various visualization techniques. We also seek to analyze user satisfaction with the explanations as

Algorithm 2 Feature selection for providing explanations

```
Inputs:
userSelectedFeatures: user selected features
shapValue: SHAP values for all features
features: list of all features
Output:
relevantFeatures: relevant feature list
Begin
    avgShapVal ← $\overline{\text{shapValue}}$
    shapFeatures ← {}
    for feat ∈ features do
        if shapValue[feat] > avgShapVal then
            shapFeatures ← feat
        end if
    end for
    relevantFeatures ← shapFeatures∪ userSelectedFeatures
    Return: relevantFeatures
End
```

well as to assess the explanations' effectiveness in the decision-making.

5. Experiment

This section describes the data sources, provides the overview of the data, elaborate on the fault types in the data set, gives descriptions on the data preprocessing and provides details on the fault diagnosis model implementation.

5.1 Data Collection and Preparation

First, let us understand our air handling unit data.

5.1.1 Description of the System

In this paper, we consider the data obtained from a shopping mall that was renovated over a decade ago. The facility has three floors that are mostly heated by the group of air handling units. The building is heated with district heating while the cooling is provided by two chillers.

Almost every large commercial building has a building management system (BMS) that contains thousands of data points that are presented through a user interface in real-time. A BMS is usually devoted to information flow and communication towards the HVAC equipment. Besides monitoring, it also provides custom reactive alarms to notify the operators at different levels. Data acquisition is accomplished through dedicated BMS in the facilities. The method for data reading and writing is the API connection supported by the BMS. Remote connection via APIs varies depending on the deployed software that each of them requires custom solutions for reliable data communication. Finally, the data transmission is secured through encrypted VPN tunnels. Data through BMS is read every 15 minutes and samples of an air handling unit were collected for the whole year in the period from February 01, 2020 to March 31, 2021. That includes measurements obtained from an air handling unit during winter, spring, summer, and autumn seasons in Estonia. In commercial buildings, AHUs usually follow an operating schedule which control the system to be switched on during occupied period and off during unoccupied mode. Before the analysis, data is filtered to exclude detected extreme outliers and samples during non-operating periods. It was further processed and faults were labeled by a dedicated HVAC engineer.

The dataset includes 10 input features as shown in Table 2. This contains samples of air handling unit under normal operating conditions and five types of faults listed in Table 3. The fault cases are taken from real scenarios and operating conditions.

5.1.2 Descriptions of Features

We observe measurements from different sub-components of the AHU, such as supply fan, heating and cooling coil valve, and heat recovery. Temperature is measured across various points in the AHU. For example, after the cooling coil, there is a temperature sensor that measures the temperature of the supply air that will enter the zones. The same applies for the rest of the temperature measurement, including the return air temperature (ARAT) and the mixed air temperature or air temperature after heat recovery (AHRST).

The heating and cooling coil valve openings, which are measured in %, are the control values sent from the BMS to open or close the valves. This is important to distinguish from the actual valve opening value since there can be the case that physical barriers causing the valves to not open or close properly, leading to malfunction in the operation of AHU although the control signal of the valves is at normal value. The supply fan static pressure (ASFPE), measured in Pascal (Pa), and the supply fan speed (ASFS), measured in %, are correlating features that indicate the working of the supply fan.

Here, the supply air temperature (ASAT) is the control variable. It is one of the main variables that indicates indoor thermal comfort. Ideally, its value should stay very close to the setpoint temperature (ASATCSP). When the setpoint is not reached, it triggers value change in other AHU variables to bring the supply air temperature close to the setpoint again.

Ambient temperature (AAT) is the main weather-related variable used in this study. Climate factor influences indoor thermal comfort. Therefore, the supply air temperature setpoint is not fixed and it is calculated depending on the value of ambient temperature. Different setpoint temperature is set in different seasons in order to allow for energy saving and other indoor thermal comfort factors.

Different factors can influence the working of the AHU. Climate conditions determine the mode of operation, i.e., heating or cooling mode. Components can degrade overtime, which leads to inefficiencies, etc. The acceptable measurement range also varies from AHU to AHU. Moreover, it is important to distinguish the steady-state mode and the transient mode. During transient-state, immediately when the ventilation machine is switched on, the measurement values show temporary fluctuation. The values shift towards the steady-state mode very quickly. However, in this study we are only looking at the steady-state operation.

Table 2. Description of the used features.

No.	Feature	Short Description	Unit
x1	AAT	Fresh air intake temperature	°C
x2	ACCVO	Cooling coil valve opening	%
x3	AHCVO	Heating coil valve opening	%
x4	AHRS	Heat recovery rotation speed	%
x5	AHRST	Supply air temperature after heat recovery	°C
x6	ARAT	Return air temperature	°C
x7	ASAT	Supply air temperature	°C
x8	ASATCSP	Supply air temperature calculated setpoint	°C
x9	ASFPE	Supply fan static pressure	Pa
x10	ASFS	Supply fan speed	%

5.1.3 Description of Faults Under Observation

Five faults representing failures in sensor, heat recovery, heating coil and cooling coil valve of the AHU are described. The faults are introduced when the system is supposed to be operating at steady-state conditions and the symptoms from dominant features that correspond to the fault have occurred. All faults described here are representation of actual fault scenarios from an air handling unit of a commercial building.

Fault 1 is a malfunction of the fan pressure sensor. The fan pressure sensor measurement is used to calculate the control value for the supply or return fan speed. The fault causes the control signal to the supply fan speed to be at an undesirable range. Thus the important features in this fault shall include the fan pressure (ASFPE) and the fan speed (ASFS).

Fault 2 is the failure of the heat recovery. During normal operation, the heat recovery should operate at 70% efficiency or above. The heat recovery is utilized before the heating coil is used. The features important in determining this fault include heat recovery speed (AHRS), return air temperature (ARAT), ambient temperature (AAT), and supply air temperature (ASAT).

Fault 3 is heating coil valve leakage. The fault indicates that the heating coil valve is not closing totally when there is a command to close it. Regardless of the fact that the valve should be closed, the hot water flows through the coil and heats up the supply air. This results in the extra heating cost, and may even lead to the extra cooling costs and undesired supply air temperature. The leak can be detected by checking the temperature sensors in the supply air channel, or comparing the work of heat recovery and cooling coil with other ventilation machines or this machine's typical actions. The important features in this fault include, heating coil valve opening (AHCVO), supply air temperature (ASAT), and supply air temperature after heat recovery (AHRST).

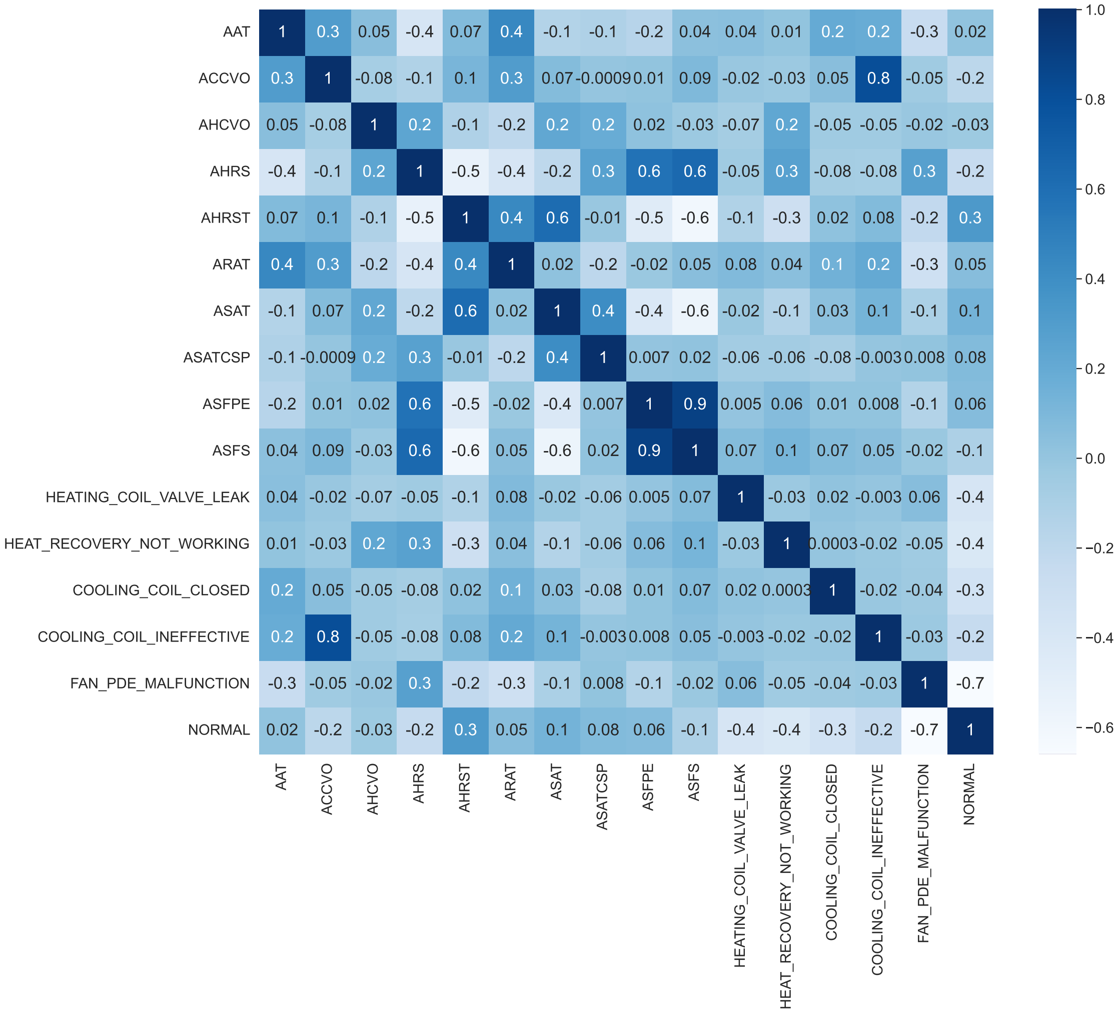
Fault 4 is the stuck cooling coil valve. The fault indicates that the cooling valve is stuck at a lower value and the ventilation unit is not fully utilizing the cooling capacity. The important features in this fault includes the cooling coil valve opening (ACCVO), supply air temperature (ASAT), and calculated supply air temperature setpoint (ASATCSP).

Fault 5 is the closed cooling coil. The fault implies that ventilation unit controller is not sending a command to fully utilize the cooling capacity. This might indicate the problem with the PID controller. The important variables in this fault type includes the cooling coil valve opening (ACCVO), supply air temperature (ASAT) and the supply air temperature setpoint (ASATCSP).

Table 3 lists the mentioned faults, with each fault corresponding to different component of the AHU. Fault-free samples are labeled as "Normal." Figure 6 shows the correlation between the features and the faults. Note that the coefficient is generated from the samples during occupied mode only. Feature pairs with correlation coefficient close to 1 indicates a positive relationship while -1 indicates a negative relationship. Coefficient close to 0 means the feature pair has no correlation at all. It can be noticed from the plot that the cooling coil valve opening (ACCVO) has strong relationship with the fault type COOLING_COIL_INEFFECTIVE. The variable heating coil valve opening (AHCVO) and heat recovery speed (AHRS) also have some positive relationship with the fault type HEAT_RECOVERY_NOT_WORKING.

Table 3. List of AHU faults used in the analysis.

No.	Abbreviation	Fault Type	Component	Sample Size
F1	FPES_M	Fan pressure sensor malfunction	Fan Pressure Sensor	894
F2	HR_NW	Heat recovery not working	Heat recovery	1146
F3	HCV_L	Heating coil valve leakage	Heating coil	794
F4	CCV_S	Cooling coil valve stuck	Cooling Valve	434
F5	CCCV	Closed cooling coil valve	Control	768
–	Normal	–	–	20925



	AAT	ACCVO	AHCVO	AHRS	AHRST	ARAT	ASAT	ASATCSP	ASFPE	ASFS	HEATING_COIL_VALVE_LEAK	HEAT_RECOVERY_NOT_WORKING	COOLING_COIL_CLOSED	COOLING_COIL_INEFFECTIVE	FAN_PDE_MALFUNCTION	NORMAL
AAT	1	0.3	0.05	-0.4	0.07	0.4	-0.1	-0.1	-0.2	0.04	0.04	0.01	0.2	0.2	-0.3	0.02
ACCVO	0.3	1	-0.08	-0.1	0.1	0.3	0.07	-0.0009	0.01	0.09	-0.02	-0.03	0.05	0.8	-0.05	-0.2
AHCVO	0.05	-0.08	1	0.2	-0.1	-0.2	0.2	0.2	0.02	-0.03	-0.07	0.2	-0.05	-0.05	-0.02	-0.03
AHRS	-0.4	-0.1	0.2	1	-0.5	-0.4	-0.2	0.3	0.6	0.6	-0.05	0.3	-0.08	-0.08	0.3	-0.2
AHRST	0.07	0.1	-0.1	-0.5	1	0.4	0.6	-0.01	-0.5	-0.6	-0.1	-0.3	0.02	0.08	-0.2	0.3
ARAT	0.4	0.3	-0.2	-0.4	0.4	1	0.02	-0.2	-0.02	0.05	0.08	0.04	0.1	0.2	-0.3	0.05
ASAT	-0.1	0.07	0.2	-0.2	0.6	0.02	1	0.4	-0.4	-0.6	-0.02	-0.1	0.03	0.1	-0.1	0.1
ASATCSP	-0.1	-0.0009	0.2	0.3	-0.01	-0.2	0.4	1	0.007	0.02	-0.06	-0.06	-0.08	-0.003	0.008	0.08
ASFPE	-0.2	0.01	0.02	0.6	-0.5	-0.02	-0.4	0.007	1	0.9	0.005	0.06	0.01	0.008	-0.1	0.06
ASFS	0.04	0.09	-0.03	0.6	-0.6	0.05	-0.6	0.02	0.9	1	0.07	0.1	0.07	0.05	-0.02	-0.1
HEATING_COIL_VALVE_LEAK	0.04	-0.02	-0.07	-0.05	-0.1	0.08	-0.02	-0.06	0.005	0.07	1	-0.03	0.02	-0.003	0.06	-0.4
HEAT_RECOVERY_NOT_WORKING	0.01	-0.03	0.2	0.3	-0.3	0.04	-0.1	-0.06	0.06	0.1	-0.03	1	0.0003	-0.02	-0.05	-0.4
COOLING_COIL_CLOSED	0.2	0.05	-0.05	-0.08	0.02	0.1	0.03	-0.08	0.01	0.07	0.02	0.0003	1	-0.02	-0.04	-0.3
COOLING_COIL_INEFFECTIVE	0.2	0.8	-0.05	-0.08	0.08	0.2	0.1	-0.003	0.008	0.05	-0.003	-0.02	-0.02	1	-0.03	-0.2
FAN_PDE_MALFUNCTION	-0.3	-0.05	-0.02	0.3	-0.2	-0.3	-0.1	0.008	-0.1	-0.02	0.06	-0.05	-0.04	-0.03	1	-0.7
NORMAL	0.02	-0.2	-0.03	-0.2	0.3	0.05	0.1	0.08	0.06	-0.1	-0.4	-0.4	-0.3	-0.2	-0.7	1

Figure 6. Correlation plot representing relationship between independent variables and the AHU faults.

5.1.4 Data Pre-processing

The data was pre-processed by removing samples during off-state. This is because AHU is not fully controlled during unoccupied mode, so the data pattern are not the same as the occupied mode. Figure 7 shows the measurement of the controlled variable supply air temperature during a 7-day period. Ventilation unit is switched on at 08:00AM and off again at 22:00PM, during the occupied period.

Since we're only interested in the steady-steate operation, the samples during first hour of operation from each day is also removed. Anomalies in the dataset were also removed, i.e., faults that appear at very few frequencies.

The input data is split into 66% and 34% for the train and test sets, respectively. Random stratified sampling is applied in the data partitioning process to keep the balance of fault classes for both sets.

Table 3 shows that the samples of normal operation (majority class) exceed those of faulty cases (minority class) with an extreme imbalance. Having such imbalanced classes for classification problems can lead to the biased predictions towards the majority class. This problem is tackled with random under-sampling techniques to transform the class distribution in the training set and eliminate the extreme data imbalance.

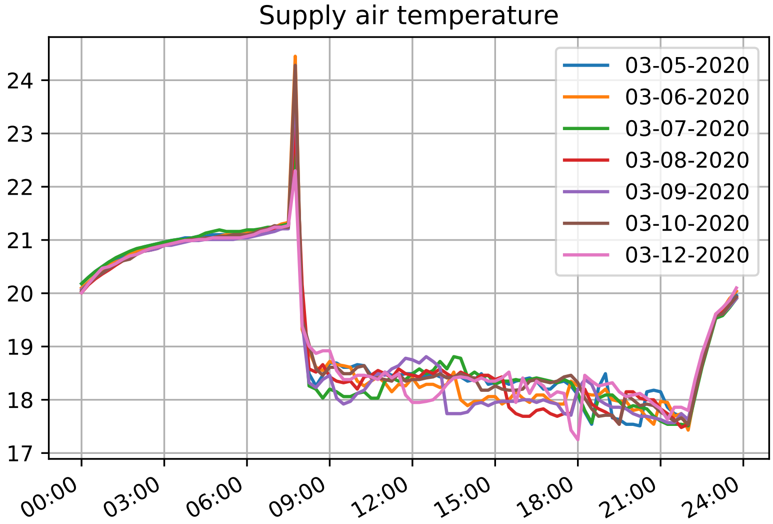


Figure 7. One-week observation of supply air temperature measurement from 03-05-2020 to 03-12-2020.

5.1.5 Feature Engineering

Using heuristic approach and simple expert rules, new features are derived. Heat recovery efficiency is the function of AHU related temperatures. It is also the indication of the amount of heat that gets recycled. It is defined in the following:

$$\text{HREfficiency} = \frac{\text{AHRST-AAT}}{\text{ARAT-AAT}}. \tag{5.1}$$

Temperature difference before and after heating coil is necessary in deriving the state of the heating coil and is defined as:

$$\text{tempDiffHC} = \text{ASAT} - \text{AHRST}. \tag{5.2}$$

Temperature difference between supply air temperature and supply air temperature setpoint is necessary in determining the effectiveness of the control and is denoted by:

$$\text{deltaSupplyTemp} = \text{ASAT} - \text{ASATCSP}. \tag{5.3}$$

Table 4. Extracted features.

No.	Feature	Short Description	Unit
x11	tempDiffHC	Temperature difference before and after heating coil	°C
x12	HREfficiency	Heat recovery efficiency	%
x13	deltaSupplyTemp	Difference between supply air temperature and supply air temperature setpoint	°C

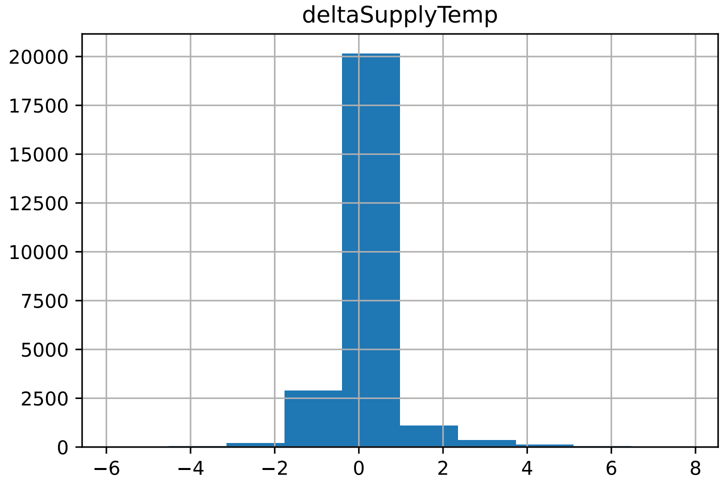


Figure 8. Histogram of variable delta supply air temperature.

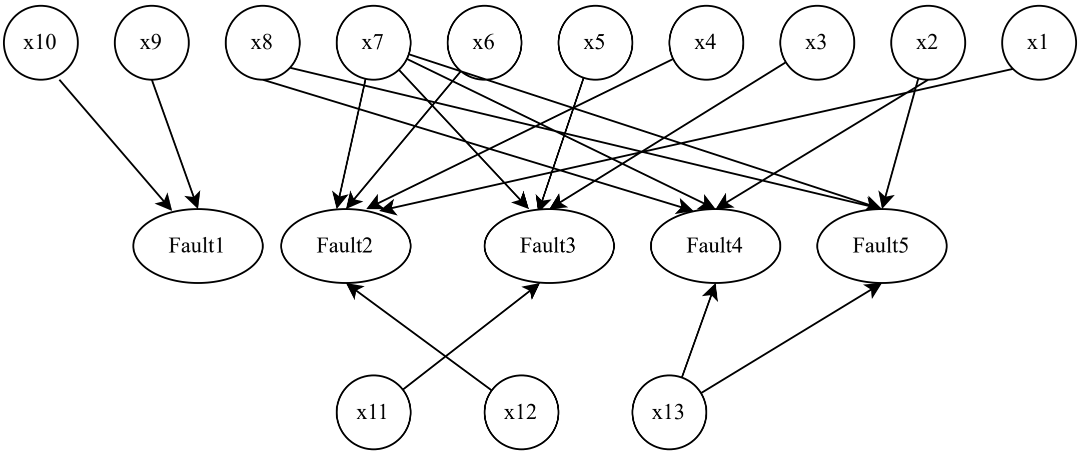


Figure 9. Feature mapping to their corresponding faults.

We now obtained new sets of features, which are summarized in Table 4. Figure 8 shows the distribution of delta supply air temperature. Most of the values range approximately between -0.5 and 1. There seems to be measurements where the difference between supply air temperature and setpoint is very high which might be either outliers or the fault of the AHU.

Based on the complete feature set, which includes original features and extracted features, it is possible to generate a feature mapping as shown in Figure 9, which maps the features to their corresponding fault types, using expert knowledge. This will serve as the crucial information in later step when communicating the fault prediction to end-users.

5.2 Implementation of the Black-Box Models

In this study, the model aims to predict whether the AHU is operating at normal or faulty condition at specific timestamps, and which fault type(s) are present. For training the fault diagnosis model, the problem is formulated as a multi-label classification problem, where the labels are binary vectors (value 0 or 1 for each of the five fault classes, plus the normal class) and more than one fault type(s) can be present simultaneously. The train set is used to train three machine learning models, including XGBoost, and the baseline models, logistic regression and random forest.

5.2.1 Baseline Model

As described in the previous section, logistic regression is a simple regression technique that can be applied for classification tasks. It can capture the relationship between dependent variables and the variable of interest. The main advantage of using this technique is that it is interpretable and can be easily understood compared to other sophisticated machine learning models.

Here, a multiclass multilabel classification model is implemented using logistic regression as the baseline model to capture the feature and fault relationship. A second baseline model is implemented using multiclass multilabel classification model with random forest. In tuning the hyperparemeters of the random forest model, the minimum number of samples in the leaf nodes, min_samples_leaf, is set to 2 and number of estimators used is 10. The baseline models will be used to compare with the XGBoost model.

5.2.2 XGBoost and Parameter Tuning

XGBoost is a more sophisticated ensemble tree method that contains a larger set of configuration space. The hyperparameter is tuned as the following: the number of estimators used is 10. In order to reduce over-fitting problem, the column subsample ratio is set to 0.9 and alpha regularization term is set to 0.005.

6. Numeric Results and Analysis

In this section, we discuss and compare the numerical results obtained from the trained models from the previous section, conduct comparative study on the high-performing models, provide case-by-case analysis for model predictions of each fault type, and cover the interview conducted with the domain experts.

6.1 Performance Results

The performance is evaluated using the test set for the trained models—logistic regression, random forest, and XGBoost. The accuracy, precision, recall, sensitivity, specificity, and F1 scores are displayed in Table 5. The XGBoost method achieves the highest overall performance for most fault types.

Table 5. Performance matrix of the used models in the fault diagnosis task.

Model	**Fault Class**	**Accuracy**	**Precision**	**Recall**	**Sensitivity**	**Specificity**	**F1**
LR	HCV_L	0.981	0.731	0.828	0.828	0.987	0.776
	HR_NW	0.995	0.932	0.971	0.971	0.996	0.951
	CCV_S	0.997	0.888	0.972	0.972	0.997	0.928
	FPES_M	0.999	0.996	1	1	0.999	0.998
	CCV_C	0.985	0.791	0.748	0.748	0.993	0.769
	Normal	0.883	0.954	0.889	0.889	0.861	0.920
	Weighted	0.859	0.943	0.900	0.900	0.893	0.887
RF	HCV_L	0.997	0.955	0.988	0.988	0.988	0.971
	HR_NW	0.999	0.997	0.997	0.997	0.999	0.997
	CCV_S	0.999	0.960	0.986	0.986	0.999	0.973
	FPES_M	0.999	0.996	1	1	0.999	0.998
	CCV_C	0.999	0.999	0.989	0.989	0.999	0.993
	Normal	0.995	0.996	0.994	0.994	0.998	0.997
	Weighted	0.993	0.996	0.994	0.994	0.998	0.994
XGB	HCV_L	0.998	0.974	0.982	0.982	0.998	0.978
	HR_NW	0.999	0.997	0.997	0.997	0.999	0.997
	CCV_S	0.999	0.973	1	1	0.999	0.986
	FPES_M	0.999	0.996	1	1	0.999	0.998
	CCV_C	1	1	1	1	1	1
	Normal	0.997	0.998	0.997	0.997	0.996	0.998
	Weighted	0.996	0.997	0.997	0.997	0.997	0.997

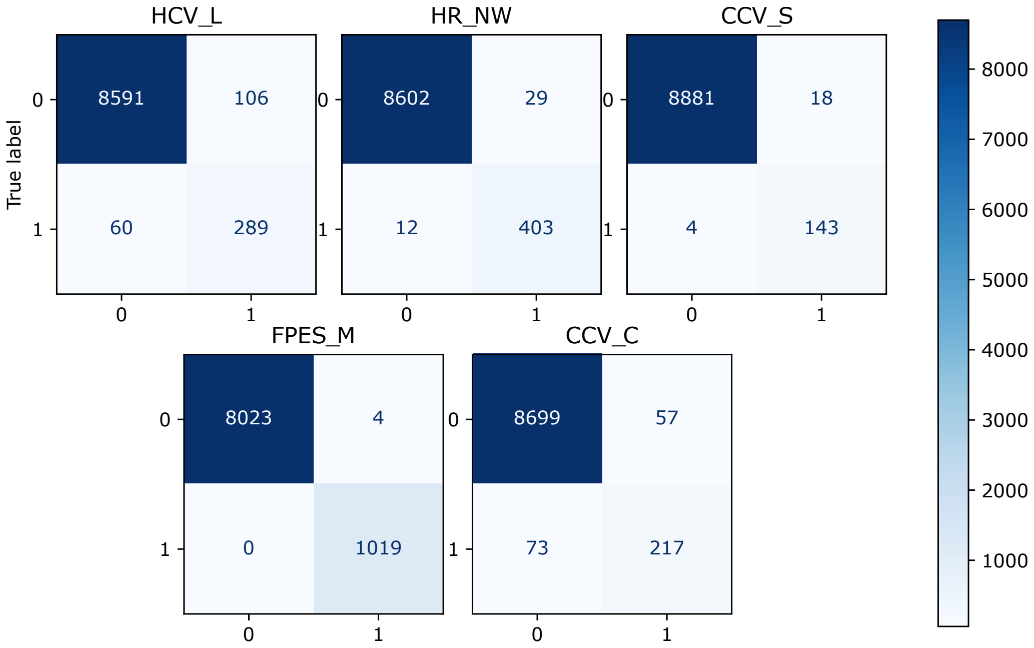


Figure 10. Confusion matrix for logistic regression model.

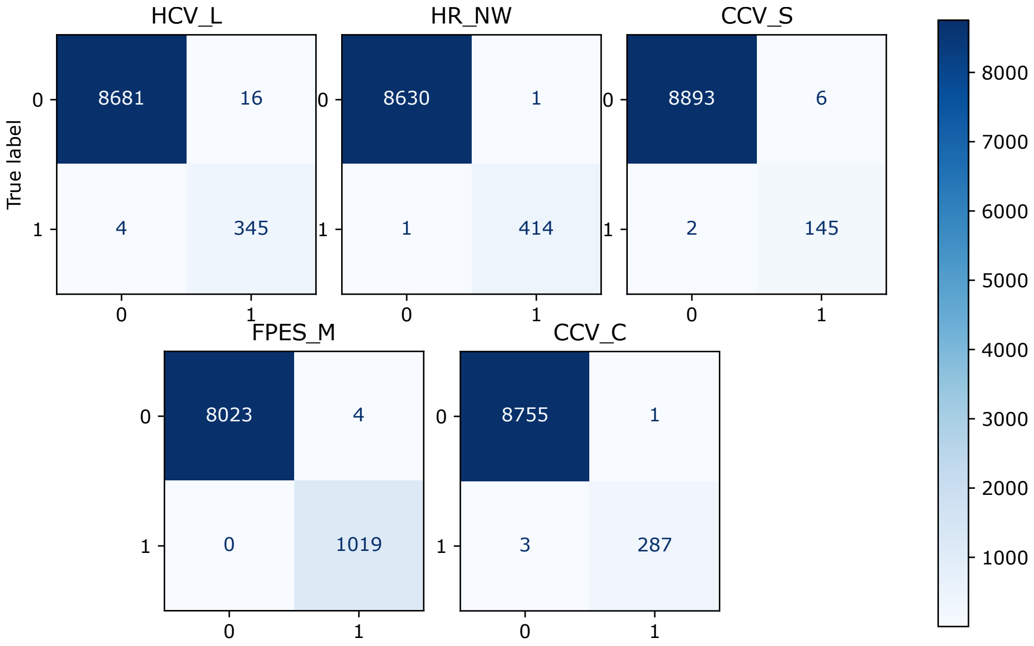


Figure 11. Confusion matrix for random forest model.

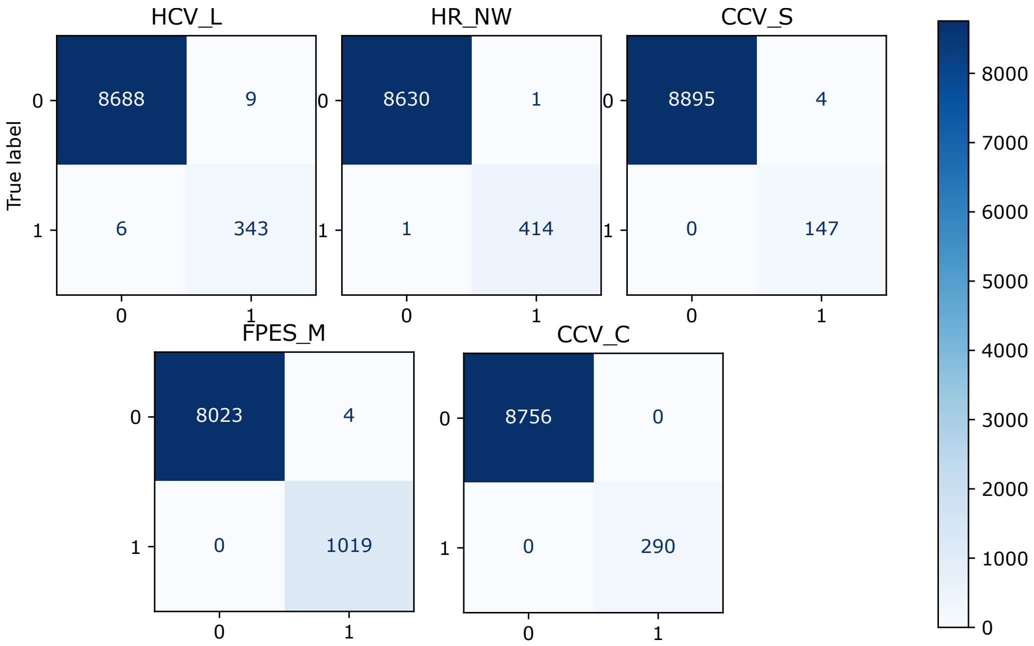


Figure 12. Confusion matrix for XGBoost model.

As Table 5 shows, the overall F1 score for logistic regression is 0.88, random forest is 0.994, and XGBoost is 0.997. Logistic regression scores 0.85 for the overall precision score, random forest and XGBoost scores 0.996 and 0.997 respectively. Logistic regression makes the prediction with 0.90 overall recall score, random forest 0.994, and XGBoost 0.997. The confusion matrix for each model prediction is shown in Table 10, 11 and 12.

6.2 Comparative Study

As shown in the confusion matrix and the performance matrix, XGBoost outperformed both of the baseline models. Random forest performance is also comparable to the XGBoost with minimal differences in the score. However, in fault detection and diagnosis tasks, it is crucial to minimize the number of false positives as much as possible. We will choose XGBoost as our fault diagnosis in this study because of the high performance. In this section, we will study the model interpretability using SHAP. The interpretations of the baseline model, random forest, and XGBoost will also be analyzed and compared.

Two types of faults are selected in this comparative study. Figures 13-16 show SHAP summary plots for XGBoost and random forest models in predicting the two types of fault classes. The features are ranked by their importance on the model prediction. The x-axis indicates the SHAP value, where a positive SHAP value means a higher contribution to the fault and a negative means a negative impact to the fault. For SHAP, the summary plot is made up of SHAP values from each individual sample. Therefore, this interpretation also represents the local interpretation. The red color indicates the high value of the variable.

The features with red colored dots most concentrated on the positive side of SHAP value axis denotes that the higher values of that features have positive influences on the fault class. The opposite applies for the blue color. If the features have blue dots concentrated on the positive side of SHAP value axis, it indicates that the low values of that features contradicts the fault instead.

Figure 13 and 14 show the SHAP summary plot of model prediction of the fault type heating coil valve leakage from the XGBoost model and random forest model, respectively. The plot shows that the first two most important variables from both models are the same, which are tempDiffHC (temperature difference before and after the heating coil) and AHRS (heat recovery speed). In the XGBoost SHAP plot, the deltaSupplyTemp (difference between supply air temperature and its setpoint) is among the top five features. Based on the plot, the higher the difference is, the more likely it is to have a heating coil valve leak. The random forest explanation also indicates the same. Random forest gives more importance to HREfficiency (heat recovery efficiency). From the domain knowledge, the heat recovery status also gives some indication of the fault heating coil valve leakage since heat recovery should be working at full speed to recycle the heat before the heating system in the heating coil should be activated. The variable AHRS (heat recovery speed), which is the second most important variable, should also correlate somewhat to the HREfficiency as well.

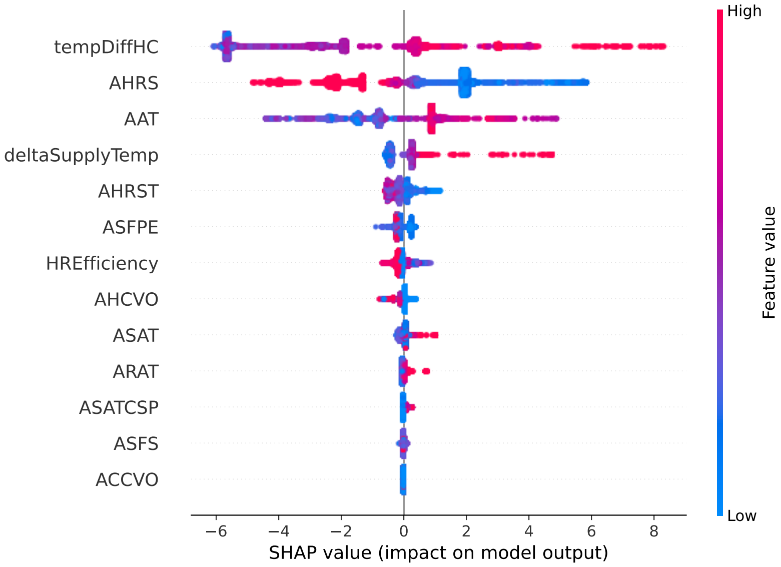


Figure 13. Summary plot of XGBoost model prediction on the fault type "Heating coil valve leak."

The SHAP summary plot of XGBoost model for predicting fault "heat recovery not working" is shown in Figure 15 and the random forest model SHAP summary plot for the same fault class is shown in Figure 16. Based on the figures, the top five most important features are the same between the two models. This includes the HREfficiency (heat

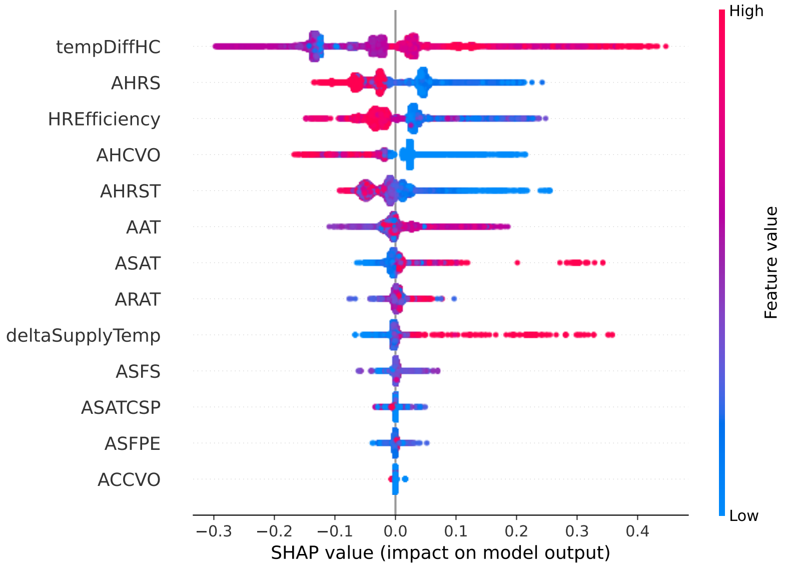


Figure 14. Summary plot of random forest model prediction on the fault type "Heating coil valve leak."

recovery efficiency), AHRS (heat recovery speed), AHCVO (heat recovery valve opening), AAT (ambient temperature), and AHRST (supply air temperature after heat recovery). There are very slight difference in the pattern, but the overall feature effects shares many similarities. For both of the models, the most important feature is the HREfficiency. Domain knowledge confirms that the HREfficiency is an important indicator of the fault malfunction of heat recovery. The rest of the features in the top five list fit in this fault description. In both summary plots, high AHRS and low HREfficiency is a sign that there is chance of malfunction in the heat recovery. One example is that the high heat recovery fan speed suggests that it is rotating but the low efficiency indicates that it fails to recycle any heat. The plots also imply that the high HREfficiency has negative influence on the fault. This maps to domain knowledge since higher efficiency means the fan is working properly.

This comparative analysis provides us insights into what is taken into account by the model in predicting the fault classes. For data with very high dimensions, it can give useful information regarding how the change in value of the top most important features can affect the model output. We have observed from our two fault examples that the generated global explanations for both models share some similar patterns and feature importance. For the first fault case, three out of the five most important features are the same between both models. And in the second fault case, five out of the five most important features are the same. Nonetheless, it is inconclusive to evaluate the models based on the XAI explanations. One of the reasons is that there is a possibility that the generated explanations might not be fully representative of the actual model. Another reason is that there can be other underlying problems such as inaccuracy in the dataset, over-fitting, or inadequacy in

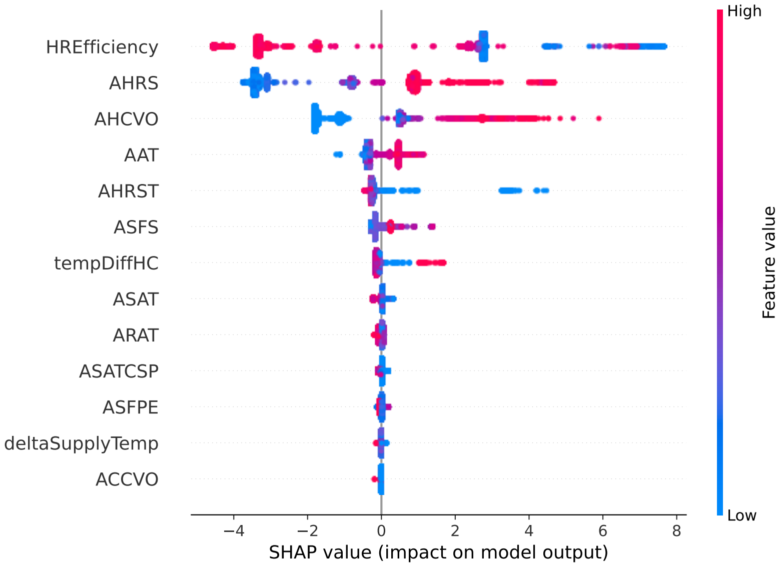


Figure 15. Summary plot of XGBoost prediction on the fault type "Heat recovery not working."

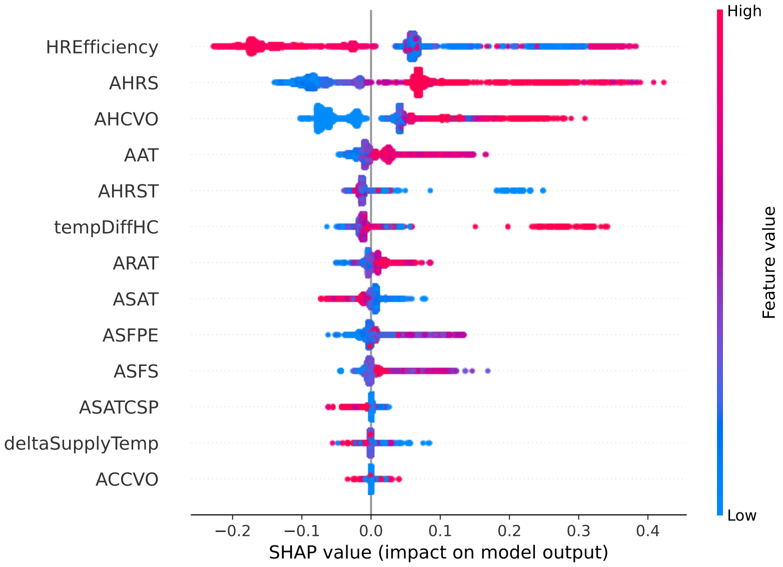


Figure 16. Summary plot of random forest model prediction on the fault type "Heat recovery not working."

the hyperparameter tuning.

6.3 Domain Requirements

We conducted a survey with seven HVAC engineers (E1-E6) who are actively working with HVAC systems. We provided a survey form which contains a list of criteria that fault diagnosis explanations may contain. The complete list can be found in the Appendix 8.1. The participants are asked to rate how important these criteria are, and they are also asked to add more criteria that they think are necessary. First, we will cover some of the feedback from the participants.

E1 and E5 are interested in viewing the fault impact in terms of the cost. More specifically, both participants mentioned that the cost factor is convincing when it comes to fault diagnosis. Knowing how many financial consequences will be created from not fixing the fault will motivate the users to take action. Other type of fault impact analysis, such as indoor climate impact, could also be a convincing factor. However, this will be regarded as part of the future work since it is out of the scope of this thesis. E2 would like to have options to choose more variables to visualize in the explanations, on top of visualizing only the most important variables. E5 would like to view the measured data in a graph with the relevant variables within one hour prior to the fault and one hour after the fault occurs. The idea is to observe when the fault probability changes and to visualize in what condition the fault is detected. E6 also thinks it's important to know how often the fault has occurred before in order to identify whether the problem is instead a result of other problems. E7 would like to see the fault-free sample to compare with the faulty sample in order to understand the expected value.

We compiled the following requirements from the feedback answers and the rating of each criteria.
R1: Option to choose variables: E2, E3, E5 and E6 would like to have the option to choose variables. E2 thinks that this option becomes less important if the result shows the most relevant variables by default. Important
R2: Visualizing the short history of faults: E1, E5, E6 and E7 think that viewing the short history of faults is important. E2 thinks that this option is different from one fault type to another. In real life, some faults occur so suddenly and in this case, the short history of fault is useful. Important
R3: Visualizing only relevant variables: E1, E2, E3, E4 and E7 would like to view only the most important features that impact the fault likelihood. In addition to **R1**, they want to view only the variables that influence the fault and have the option to select more variables. Critical

R4: Visualizing each feature attribution to the fault: E2, E4 and E5 think it's good to view in terms of probability how much each feature affects the fault likelihood. In addition, E2 also specifies that the probability of the diagnosed fault is also a convincing factor for users. For example, if the probability shows 1.00, then it is more convincing than when the probability is only 0.97. E3 and E4 don't think it's very important. E5 would like to observe the probability changes when the fault occurs. Optional

The Motivation for Enabling User-Selected Features

Figure 17 shows the prediction for one fault sample that is related to the heating coil valve leakage.

Time: "03-25 15:45"
Predicted Fault: Heating coil valve leak
Fault Probability: 1.00

Although the fault is predicted with perfect confidence, there is a critical piece of information that is not obviously visible on the plot. HVAC engineers confirm that the plot has enough information to be able to deduce a heating coil valve leakage just by looking at the single instance. In this case, there are some feature correlations that are possible to deduce from the domain knowledge, but it is not straightforward. In the ventilation unit, there is a sequence of operations where each component interacts with each other. Thus, the value of one variable may affect the other, and it may become obvious when there is some anomaly in the pattern.

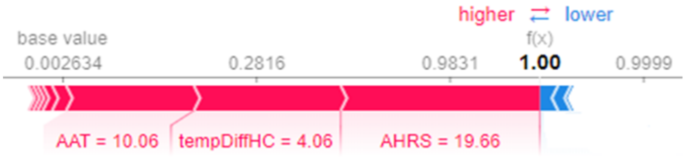


Figure 17. (Heating coil valve leakage) standard SHAP plot for individual explanations.

In the example above, the fault can be deduced based on the control sequence logic. The heat recovery must always be fully utilized before the heating coil is used. In this case, the heat recovery is not fully utilized since its rotation speed (AHRS) is only about 19%, and thus it is likely that the heating coil valve is still closed. When the heating coil is not utilized, there shouldn't be much difference in the temperature before and after the heating coil, and it should be approximately equal. From the example, the difference in the temperature (tempDiffHC) of four degrees is enough to indicate that there is excess heating consumption that is generated before the heat recovery is fully utilized. This logic applies in this specific case.

However, the information about the heating coil valve opening (AHCVO) would be crucial to help in confirming the fault. This makes it necessary to use one set of variables for one fault and other sets of variables for other fault types. And all the relevant information should be clearly visible from the visualization. It may happen in fault cases where the faults can be derived from the sequence of operations and control logic, but the relationship may not be straightforward. Therefore, it becomes crucial to provide meaningful sets of features that confirm that the faults exist.

6.4 Explaining Faults

In this section, SHAP is used as the explainable method. Explanations are provided for the XGBoost model output of randomly chosen samples from each type of fault from the dataset. The visualization design is based on the domain requirements collected from the previous sections. The figures below visualize the explanations in a sliding window format. Instead of plotting the SHAP values, actual values are provided. To show the feature attribution to the fault, the SHAP value for each relevant feature is converted to a percentage and is added in the annotation. Fault probability is also provided for the data instance where the user wants explanations for. The areas where a fault is present are highlighted with a light red background to allow for an understanding of where the fault begins.

6.4.1 Case 1: Fan Pressure Sensor Malfunction

Figure 18 shows the explanation for the fault "Fan pressure sensor malfunction" of the observed sample, which is at 03-11 18:00. To understand the progress history of the fault, the measurement values for a number of time steps, starting from 03-11 10:00, are shown. In this observed data instance, the variables ASFS (supply fan speed) and ASFPE (supply fan pressure) have a positive impact on the fault prediction. The supply fan speed is 30% and the supply fan pressure is at 3.59 Pa during the operation hour. The fault has already started multiple timesteps prior to the observed sample. Before the fault occurs, the fan and fan pressure measurements, which are 75% and 44.51 Pa respectively, show normal operation up until the values of both variables suddenly drop to very low values, which indicate the fan is barely operating. The low value of the ASFS contributes 68.26% to the fault probability and the ASFPE contributes 12.9%. Based on the domain knowledge, the fan pressure and fan speed are two correlating features where the pressure measurement is a variable used for calculating the air volumes. Therefore, the malfunction of the pressure sensor may cause failure in the air volume control as well.

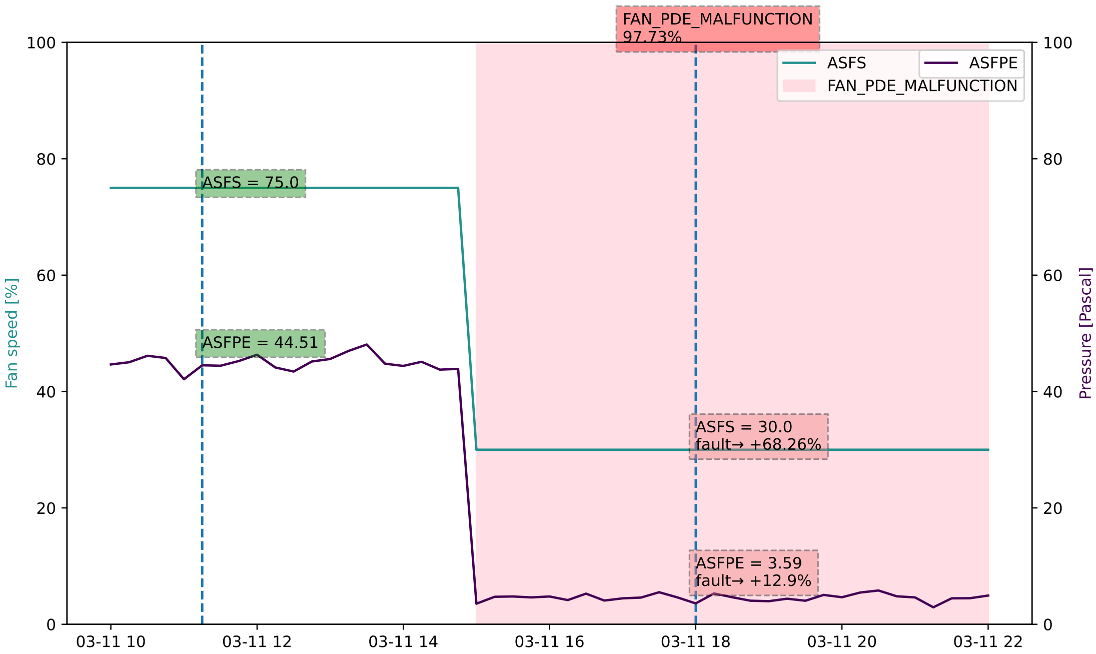


Figure 18. Explanation visualized as a sliding window for the fault type "Fan pressure sensor malfunction."

6.4.2 Case 2: Heat Recovery Not Working

Figure 19 shows samples for diagnosis of the heat recovery fault. The fault becomes very likely when the AHRS (heat recovery fan speed) is 100% and the HREfficiency (heat recovery efficiency) is only 0.01%, meaning that it fails to recover heat from the return air to heat up the supply air. When the air is not heated enough by the heat recovery unit, the heating coil valve opens to produce the heat. The AHRST (air temperature after heat recovery) is very low at that point, although the heat recovery is working at full speed. The heat recovery efficiency contributes about 16% to the fault, the heat recovery speed contributes 12%, the air temperature after heat recovery gives 26% and the temperature difference before and after the heating coil influences 13%.

6.4.3 Case 3: Heating Coil Valve Leakage

Figure 20 represents the explanation for the fault "Heating coil valve leakage". The heating coil opens for some time, from 09:00 to 11:45, to heat up the supply air and closes again. However, the temperature difference before and after the heating coil (tempDiffHC) increases, although the heating coil valve is 0. The heat recovery rotation is not utilized at 100% when the fault occurs. It could also indicate that the supply air is sufficiently heated and does not require much heat to be recycled. The low heat recovery rotation speed adds 27% to the fault. The high temperature difference before and after the heating coil adds

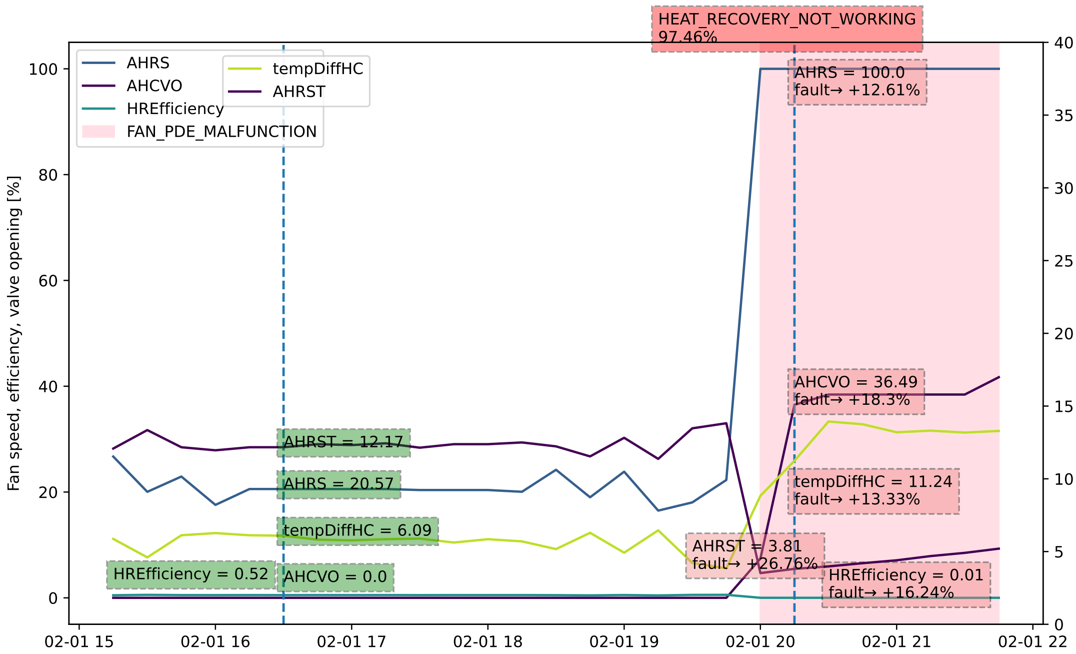


Figure 19. Explanation visualized as a sliding window for the fault type "Heat recovery not working."

another 50%. Through the domain knowledge, the information is enough to determine the fault.

6.4.4 Case 4: Cooling Coil Valve Stuck

Figure 21 shows the explanation for the fault in the cooling coil valve stuck. When the supply air temperature becomes higher than the setpoint at 19:00, the cooling coil valve opens in order to cool down the supply air. The cooling coil valve control signal shows 100%. However, the supply air temperature stays the same. And by the time of observation at 21:30, the supply air temperature is 5.13°C higher than the setpoint. The high supply air temperature adds 7.57% to the fault. The difference between supply air temperature and setpoint temperature (deltaSupplyTemp) contributes 29%, and because the cooling coil valve is 100%, the fault probability increases 53.64% more. In contrast to the non-faulty sample shown at 12:45, the deltaSupplyTemp is only 0.11 which means the supply air temperature achieves the setpoint.

6.4.5 Case 5: Cooling Coil Valve Closed

The example of a cooling coil valve closed is given in Figure 22. From the graph, the temperature difference between supply air and its setpoint (deltaSupplyTemp) increases as the ambient temperature (AAT) increases. The supply air temperature (ASAT) becomes

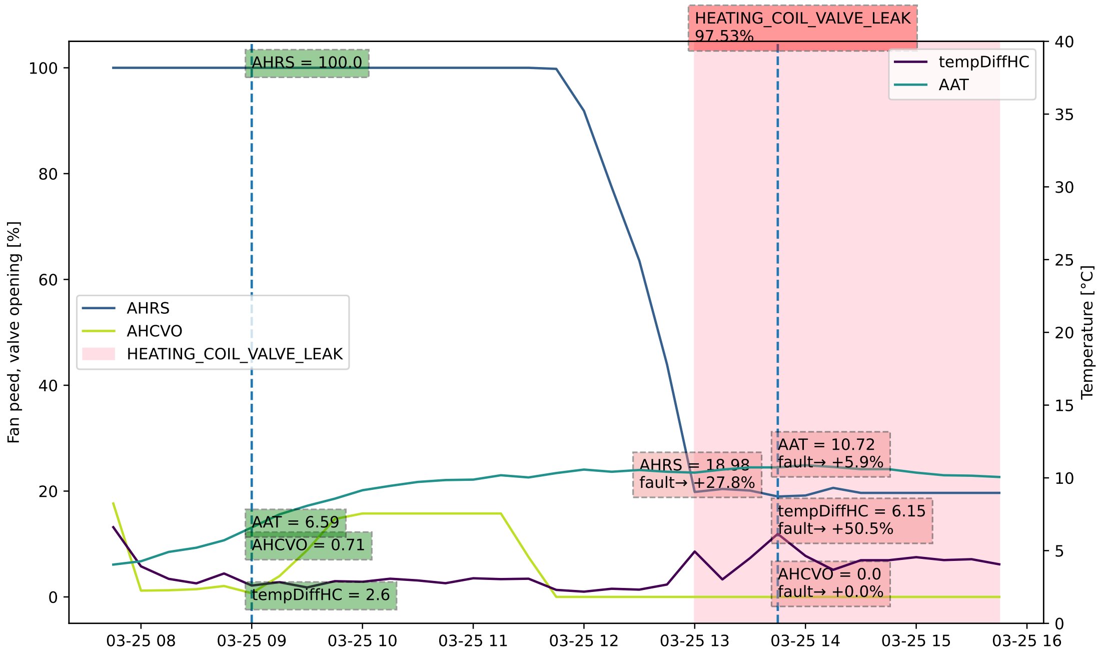


Figure 20. Explanation visualized as a sliding window for the fault type "Heating coil valve leak."

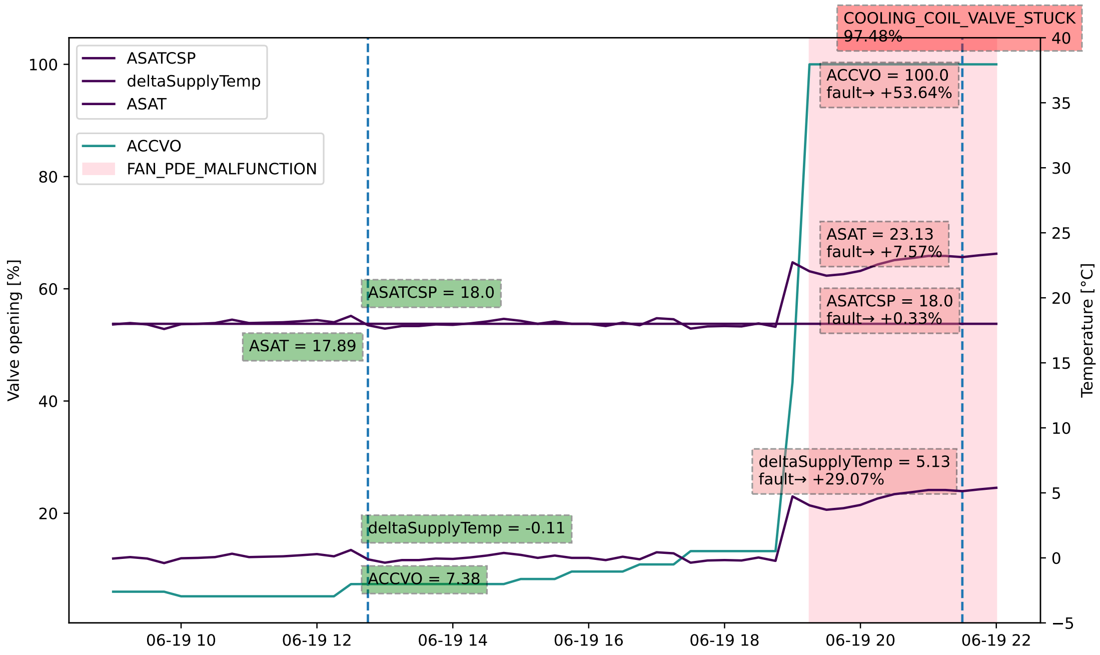


Figure 21. Explanation visualized as a sliding window for the fault type "Cooling coil valve stuck."

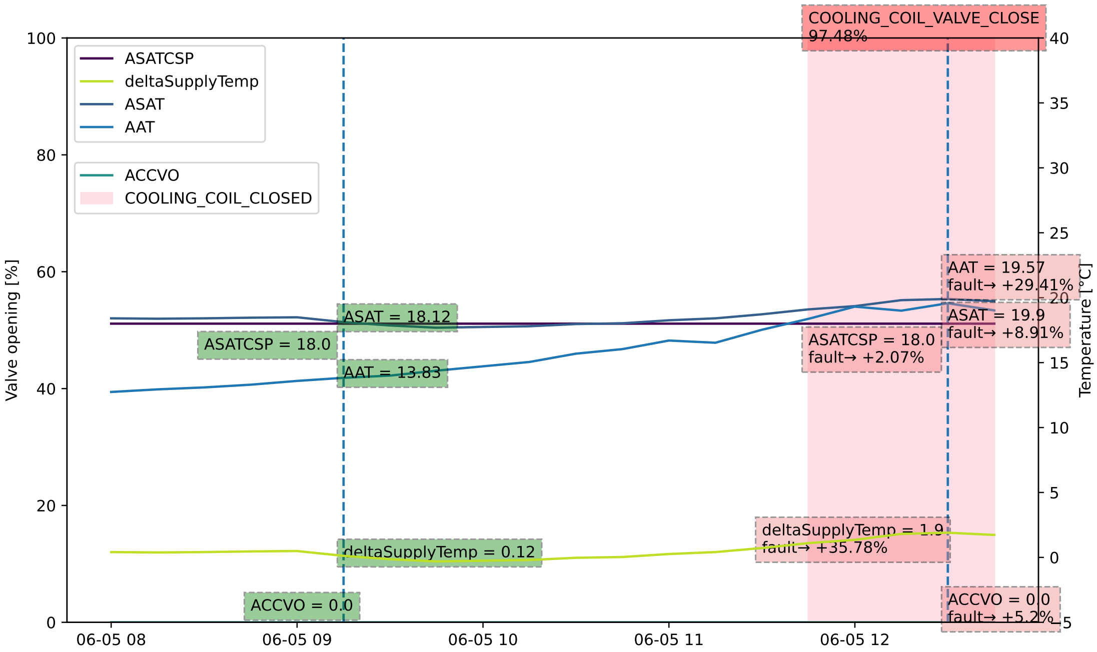


Figure 22. Explanation visualized as a sliding window for the fault type "Cooling coil valve closed."

higher. The system should send the command to open the cooling coil valve and regulate the supply air temperature to the setpoint level. However, the valve remains closed for some time-steps, which could indicate there can be problem with the BMS sending command. The increasing deltaSupplyTemp causes the fault to be approximately 35% more likely. Because the cooling valve is closed, it contributes another 5%. Ambient temperature adds to the "cooling coil valve closed" type of fault. That could be from the correlation that the ambient temperature has on when the cooling coil should be utilized.

6.5 Expert Survey

We conducted a survey with seven HVAC engineers, who also previously provided input about the domain requirements, to evaluate the fault diagnosis explanations and visualization techniques. In the experiment, experts were shown three different types of graphical representations for fault explanations.

(a) standard SHAP plot for individual instance,
(b) standard SHAP stacked plot for a specific time period, and
(c) modified version of SHAP explanation.

The experiment is conducted as a written survey. The complete questions can be found in the Appendix 8.1. Two types of faults are tested, which are "Fan pressure sensor malfunction" and "Heating coil valve leak."

First, the SHAP standard plot for individual instance is shown to explain one fault sample. Then, the SHAP standard plot for explaining multiple instances is shown for the similar fault sample. And then our version of the SHAP explanation is provided where the relevant sets of features are shown. The participants were asked to confirm if the explanations represent the actual fault and rate the satisfaction with each type of visualization techniques. They were also asked what the important criteria were to help in the fault diagnosis decision making process.

EXPERT INSIGHTS

Explanation assessment:

In general, the participants rated the modified SHAP explanation plots (c) the highest, and they also showed the most satisfaction with the visual representation from (c). Table 6 shows the explanability score and user satisfaction score obtained from the survey. The mean and median ratings are provided. Since users were asked to rate the explanations using a 5-point Likert scale, the explanability score was converted from the scale into values from 1 to 5 to obtain a numeric metric. For both fault types, participants on average agree more to the explanation (c) than they do with (a) and (b). For user satisfaction, explanation (c) ranked higher than (a) and (b) on average and in both of the fault cases.

Table 6. Evaluation results for three different types of explanation visualizations for two types of faults.

		FPES_M			**HCV_L**		
	Measures	**(a)**	**(b)**	**(c)**	**(a)**	**(b)**	**(c)**
Explanability score	mean	3	2.71	3.71	3.14	2.28	4.71
	median	4	2	5	4	1	5
User satisfaction	mean	5.71	6.57	9	5.71	5.85	7.85
	median	6	7	9	6	6	9

Expert Feedback:

Through the numeric results obtained from the previous section, participants also have extra comments to help improving the result.

- E2 thinks that different types of faults should have their own ways of representations. Faults are different in terms of how they develop. As an example, the heating coil fault is the type of fault that appears gradually. Therefore, an explanation for one single time instance may be enough to determine the fault at the current state. The opposite case applies to faults that occur very suddenly. For example, in the case of the malfunction of the fan pressure sensor, the fault appears abruptly, thus showing a short history of the fault helps to identify the problem. In the fan pressure sensor failure case, the fan pressure measurement drops so suddenly that the users are able to see the obvious change in the pattern where the fan speed also drops very low. In comparison, the faults that develop gradually may not always be obvious to notice from only a short history.

- E1 would like further improvement in the visualization. The participant also mentions that the plot contains a lot of visual noise. That is not a problem with measurement values that are more static. But it may become a problem if there are too many values that change over the short visualization history. Currently, all the annotation labels for each variable are highlighted with the same color, either green or red color. Therefore, more colors would help to distinguish between different variables and make it easier to follow.
- E5 gives positive feedback, but would also like to see more related variables that would help to validate the impact of the fault. The cost impact, thermal comfort impact, and the component lifespan impact would help in understanding the importance of the fault and allow the HVAC engineers to prioritize the maintenance activities accordingly.
- E4 commented on the fault type fan pressure sensor malfunction that the features shown are relevant. But the participants pointed out that the supply fan pressure should receive more weight than the supply fan speed itself, which is the opposite in the explanation graph. However, it is the auto-generated feature importance from the model itself or it may be the representation of the samples in the dataset. This is important as user can immediately determine whether to trust the system. If the model gives correct weight to correct and relevant features, then it will build trust more.
- E6 commented on the heating coil valve leakage fault that the explanations given are sufficient but there can be further uncertainties. It is difficult to say exactly that it is the fault in the heating coil since there can also be design problems that cause the temperature increase. It may be that the temperature measurement locations are not exactly before and after the coil, therefore the temperature increase may instead be due to the heat transfer through the long ducts. For the fault type fan pressure malfunction, E6 was also uncertain after looking at the short history and would like to view the history from the previous day as well to understand if it is caused by the schedule of the building.

Analysis of the Explanation:

For both types of faults in the questionnaire, when presented with plot (c) and asked to assess the explanability, most participants picked the answer 'Strongly Agree' while they pick 'Neutral' or 'Somewhat Agree' to plots (a) and (b). After further analysis of the participants' feedback from the survey, we have summed up the findings into main points as following:

1. Users would like to visualize relevant features. This includes the most important features impacting the fault and the features that would further aid in confirming or rejecting the fault.
2. Users want to see how frequently the fault has appeared. Therefore, more flexibility in the sliding window graph is required, i.e., an option to view the short history from the day before the fault occurs or even earlier.
3. Users want to have a reference point, i.e., expected values, which they can compare with the values of the faulty sample.

7. Discussion

In this paper, we presented an approach for explaining some common air handling unit faults identified by an XGBoost Classifier using SHAP. We demonstrated the potential of SHAP as an explainable method to aid in user decision making in the area of building energy systems. We adopted XAI into the data-driven fault detection and diagnosis pipeline and explored different ways to communicate the faults to end-users, who are the HVAC experts. We defined relevant sets of features for each type of faults using the domain knowledge. Then, we integrated pre-defined features in the explanation visualization and removed features with low SHAP values to reduce the visual noise. We learnt from a survey with domain experts that some faults require understanding of what has happened at earlier time steps to help in identifying the problems. In that regard, the fault diagnosis is explained based on sliding window observations. Then, we evaluated the influence of the generated explanation on the users' decision making as compared to the standard SHAP explanation plots. The survey findings have indicated that, on average, the confidence that users have on the fault diagnosis has shown to improve, so does the user satisfaction with the generated explanations. In our work, we integrated domain knowledge to guide the design of the fault detection and diagnosis pipeline and to analyze the feature attributions of the model for each type of fault. This provides us further insights into how the fault diagnosis information is perceived by the HVAC engineers.

7.1 Limitations

This paper presents an approach to communicate the fault explanations to the end-users in the HVAC domain to help build trust in the decision making. However, this study has limitations listed in the following:

1. The current work is limited to the dataset from a single air handling unit. It may be possible to use the model to diagnose the specific AHU used to train the model. However, aggregating datasets from multiple AHUs would be useful in generalizing the faults, which would also help in studying the practicality of the data-driven method for large-scale fault diagnosis systems. Then its usage won't be restrained to only one machine.
2. Since we used the dataset from a single AHU, the work is also restrained to only

the fault types that have occurred in that specific AHU. Therefore, having more data from more AHUs would help adding more diverse sets of faults. This would be useful in terms of scalability as the model would be able to detect more fault types in more AHUs. It would provide further insights as more interesting fault types are explored.

3. Related to the previous point, our dataset contains only the minimum sets of variables. As mentioned in the introduction section, building owners tend to install the minimum number of sensors necessary for control. Because of the costs, fault detection tasks requires more financial justification of why that extra costs is necessary. For some ventilation machines, not all the sensors required for fault detection tasks are available. Features such as meter values, that may provide valuable information on the heating coil valve leakage fault, are not always measured. Although the faults can be derived based on the fault symptoms that appear, it is still difficult to pinpoint under full certainty what are the causes of the fault in the case that crucial sensor is missing. Thus it also limits the explanation capability and users still aren't convinced of the predicted fault after provided with the explanations.
4. The scope of the result analysis was limited to only some basic scenarios of faults that can be observed from the selected examples. In real life, the interaction between sub-components in AHU is complex and the fault behavior varies from one AHU to another. Different AHUs are also different in terms of the design architecture and how the sensors are placed that may influence the end result of how the faults are perceived.
5. This work is only limited to one type of sensor fault, which is available in the dataset. Sensor faults are very common for the building's systems. Examples may include sensor bias and sensor drift. This problem may cause uncertainties in the fault diagnosis since the system may read the wrong measurement and classify the samples as anomalies. Therefore, identifying sensor faults would mitigate the probability of giving the wrong diagnosis and false alarms. Further analysis of sensor faults is crucial for the fault diagnosis task.
6. The evaluation of the fault explanations is constrained only seven participants. The result should be validated with a bigger sample size to help producing statistically meaningful tests results.
7. The XAI technique used in this work is restrained to only one method, SHAP. Because of time constraint, we selected only one explainer method and explored the use cases of it and study the domain requirements from the end-users. Ideally, the study should incorporate more explainers, such as LIME and CIU. It would be useful to analyze how different XAI techniques differ and observe the user satisfaction between explanations from each type of explainer.
8. In this work, the definition of a good explanation is limited to only the scores

from user evaluation. However, a more systematic approach would produce a more accurate evaluation and make the tasks less manual. Human users may make mistakes in the evaluation process, and one user may also perceive the fault explanations differently from another users.

9. This work is limited to only the model-specific version of SHAP, which is TreeSHAP. The domain requirement in this work is geared towards industrial scale system. Therefore, KernelSHAP is not considered in our case. It is important to take into account the run time of generating the explanations as the fault detection task is in real-time. However, the real-time application of the explainable method also needs to be tested further. Theoretically, TreeSHAP is fast and applicable for practical tasks, which may not cause any major problems. But depending on the domain requirements, the computation speed may get factored in when deciding whether the explanable method should be enabled for fault monitoring.
10. We covered briefly the comparison between explanations from the XGBoost model and the baseline RandomForest model. Further work may be required in order to assess the explanability more systematically to see which model provide better overall representation of the faults.

8. Conclusion and Future Work

Advanced machine learning techniques have recently demonstrated excellent performance in fault detection and diagnosis problems. Nevertheless, building personnel may find it hard to evaluate and understand the reasoning of the produced outputs. In this light, we developed an approach that utilizes the game theory-based SHAP method to explain the output of an XGBoost classifier for fault detection and diagnosis tasks. We presented the method for explaining the relevant features as a sliding window analysis. The obtained results are validated by the HVAC engineers. This idea is demonstrated using real data collected from a commercial building.

8.1 Future Work

From the research findings and limitations addressed in the previous sections, the following research directions may be considered for future studies:

1. The explanation and visualization methods should be improved further to cover more types of faults in a more extended dataset. For example, different types of faults, such as gradual faults and abrupt faults, may require different methods to be explained. For some faults, a longer history should be visualized, while some is sufficient with only an individual instance explanation. More extensive involvement from HVAC engineers is necessary in order to provide explanations that suit various types of faults and are effective in communicating information to the end-users.
2. One possible future direction is to conduct an experiment and compare the explanations of different explainer methods such as SHAP, LIME, and CIU. The comparison may include criteria such as computation speed and time spent on understanding the explanation. This would help in the selection of a suitable explainable method depending on the requirements.
3. We aim to strengthen the definition of a good explanation for the fault detection of building HVAC systems. On top of user evaluation, we should adopt a further systematic approach that is less reliant on manual evaluation.
4. In order to test the scalability of the fault diagnosis method, a larger dataset is required. Dataset may include samples from AHU of different buildings and of different climate conditions. Other HVAC components, such as chillers or heat

pumps, may also be explored. Ideally, more types of faults are covered, including sensor faults and component faults.

5. For further studies, the survey should be conducted with a larger sample size that could help the results to be statistically meaningful.
6. Other challenging possibilities also include the development of an unsupervised model for fault detection tasks and applying an explainable method to understand the output.
7. We aim to expand the research to include the fault impact analysis element, which may help to provide better insights into not only what is happening but also what will happen if no action is taken on the fault. This may require development of prediction models. As an example, thermal comfort impact analysis may require forecasting of the increase in supply air temperature one hour or more following the fault. Similarly, cost impact analysis may require forecasting of energy consumption over a specific period.

Bibliography

[1] T. Abergel, B. Dean, J. Dulac, I. Hamilton, and T. Wheeler, “2018 Global Status Report - Towards a zero-emission, efficient and resilient buildings and construction sector,” Global Alliance for Buildings and Construction, Tech. Rep., 2018, [Online] Available `https://www.worldgbc.org/news-media/2018-global-status-report-towards-zero-emission-efficient-and-resilient-buildings-and`, Accessed January 24, 2022.

[2] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[3] F. Xiao and S. Wang, “Progress and methodologies of lifecycle commissioning of HVAC systems to enhance building sustainability,” *Renewable and Sustainable Energy Reviews*, vol. 13, no. 5, pp. 1144–1149, 2009.

[4] M. S. Mirnaghi and F. Haghighat, “Fault detection and diagnosis of large-scale HVAC systems in buildings using data-driven methods: A comprehensive review,” *Energy and Buildings*, p. 110 492, 2020.

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

[6] S. Srinivasan, P. Arjunan, B. Jin, A. L. Sangiovanni-Vincentelli, Z. Sultan, and K. Poolla, “Explainable AI for chiller fault-detection systems: Gaining human trust,” *Computer*, vol. 54, no. 10, pp. 60–68, 2021.

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

[8] Z. Du and X. Jin, “Multiple faults diagnosis for sensors in air handling unit using Fisher discriminant analysis,” *Energy Conversion and Management*, vol. 49, no. 12, pp. 3654–3665, 2008.

[9] H. Liao, W. Cai, F. Cheng, S. Dubey, and P. B. Rajesh, “An online data-driven fault diagnosis method for air handling units by rule and convolutional neural networks,” *Sensors*, vol. 21, no. 13, p. 4358, 2021.

[10] B. Fan, Z. Du, X. Jin, X. Yang, and Y. Guo, "A hybrid FDD strategy for local system of AHU based on artificial neural network and wavelet analysis," *Building and environment*, vol. 45, no. 12, pp. 2698–2708, 2010.

[11] Z. Du, B. Fan, X. Jin, and J. Chi, "Fault detection and diagnosis for buildings and HVAC systems using combined neural networks and subtractive clustering analysis," *Building and Environment*, vol. 73, pp. 1–11, 2014.

[12] H. Han, B. Gu, Y. Hong, and J. Kang, "Automated FDD of multiple-simultaneous faults (MSF) and the application to building chillers," *Energy and Buildings*, vol. 43, no. 9, pp. 2524–2532, 2011.

[13] Z. Du, B. Fan, J. Chi, and X. Jin, "Sensor fault detection and its efficiency analysis in air handling unit using the combined neural networks," *Energy and Buildings*, vol. 72, pp. 157–166, 2014.

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

[15] D. Gunning, "Explainable artificial intelligence (XAI)," Defense Advanced Research Projects Agency, Tech. Rep., 2017, [Online] Available `https://www.darpa.mil/program/explainable-artificial-intelligence`, Accessed January 27, 2022. [Online]. Available: `https://www.darpa.mil/program/explainable-artificial-intelligence`.

[16] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[17] G. Joshi, R. Walambe, and K. Kotecha, "A review on explainability in multimodal deep neural nets," *IEEE Access*, vol. 9, pp. 59 800–59 821, 2021.

[18] R. Machlev, M. Perl, J. Belikov, K. Y. Levy, and Y. Levron, "Measuring explainability and trustworthiness of power quality disturbances classifiers using explainable artificial intelligence (XAI)," *IEEE Transactions on Industrial Informatics*, 2021.

[19] C. J. Hoofnagle, B. van der Sloot, and F. Z. Borgesius, "The european union general data protection regulation: What it is and what it means," *Information & Communications Technology Law*, vol. 28, no. 1, pp. 65–98, 2019.

[20] G. Li, Q. Yao, C. Fan, *et al.*, "An explainable one-dimensional convolutional neural networks based fault diagnosis method for building heating, ventilation and air conditioning systems," *Building and Environment*, vol. 203, p. 108 057, 2021.

[21] M. Madhikermi, A. K. Malhi, and K. Främling, "Explainable artificial intelligence based heat recycler fault detection in air handling unit," in *Lecture Notes in Computer Science*, Springer International Publishing, 2019, pp. 110–125.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, “" why should i trust you?" explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[23] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Advances in neural information processing systems*, vol. 30, 2017.

[24] Y. Himeur, A. Alsalemi, A. Al-Kababji, *et al.*, “A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects,” *Information Fusion*, vol. 72, pp. 1–21, 2021.

[25] J. M. House, H. Vaezi-Nejad, and J. M. Whitcomb, “An expert rule set for fault detection in air-handling units/discussion,” *Ashrae Transactions*, vol. 107, p. 858, 2001.

[26] J. Schein and S. T. Bushby, “A hierarchical rule-based fault detection and diagnostic method for hvac systems,” *Hvac&r Research*, vol. 12, no. 1, pp. 111–125, 2006.

[27] S. M. Namburu, M. S. A. S, J. Luo, K. Choi, and K. R. Pattipati, “Data-driven modeling, fault diagnosis and optimal sensor selection for HVAC chillers,” *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pp. 469–473, 2007.

[28] G. Li, Q. Yao, C. Fan, *et al.*, “An explainable one-dimensional convolutional neural networks based fault diagnosis method for building heating, ventilation and air conditioning systems,” *Building and Environment*, p. 108 057, 2021.

[29] D. Chakraborty, A. Alam, S. Chaudhuri, H. Başağaoğlu, T. Sulbaran, and S. Langar, “Scenario-based prediction of climate change impacts on building cooling energy consumption with explainable artificial intelligence,” *Applied Energy*, vol. 291, p. 116 807, 2021.

[30] S. Wenninger, C. Kaymakci, and C. Wiethe, “Explainable long-term building energy consumption prediction using QLattice,” *Applied Energy*, vol. 308, p. 118 300, 2022.

[31] M. Kim, J.-A. Jun, Y. Song, and C. S. Pyo, “Explanation for building energy prediction,” in *International Conference on Information and Communication Technology Convergence*, 2020, pp. 1168–1170.

[32] Y. Gao and Y. Ruan, “Interpretable deep learning model for building energy consumption prediction based on attention mechanism,” *Energy and Buildings*, vol. 252, p. 111 379, 2021.

[33] A. Li, F. Xiao, C. Zhang, and C. Fan, “Attention-based interpretable neural network for building cooling load prediction,” *Applied Energy*, vol. 299, p. 117 238, 2021.

[34] P. Arjunan, K. Poolla, and C. Miller, "EnergyStar++: Towards more accurate and explanatory building energy benchmarking," *Applied Energy*, vol. 276, p. 115 413, 2020.

[35] T. Tsoka, X. Ye, Y. Chen, D. Gong, and X. Xia, "Building energy performance certificate labelling classification based on explainable artificial intelligence," in *Neural Computing for Advanced Applications*, Springer Singapore, 2021, pp. 181–196.

[36] C. Miller, "What's in the box?! Towards explainable machine learning applied to non-residential building smart meter classification," *Energy and Buildings*, vol. 199, pp. 523–536, 2019.

[37] É. Houzé, J.-L. Dessalles, A. Diaconescu, D. Menga, and M. Schumann, "A decentralized explanatory system for intelligent cyber-physical systems," in *Lecture Notes in Networks and Systems*, Springer International Publishing, 2021, pp. 719–738.

[38] D. Luckey, H. Fritz, D. Legatiuk, K. Dragos, and K. Smarsly, "Artificial intelligence techniques for smart city applications," in *Lecture Notes in Civil Engineering*, Springer International Publishing, 2020, pp. 3–15.

[39] Y. G. Akhlaghi, K. Aslansefat, X. Zhao, *et al.*, "Hourly performance forecast of a dew point cooler using explainable Artificial Intelligence and evolutionary optimisations by 2050," *Applied Energy*, vol. 281, p. 116 062, 2021.

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

[41] C. Molnar, *Interpretable Machine Learning, A Guide for Making Black Box Models Explainable*, 2nd ed. 2022. [Online]. Available: `https://christophm.github.io/interpretable-ml-book`.

[42] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[43] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[44] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[45] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[46] S. Anjomshoae, K. Främling, and A. Najjar, “Explanations of black-box model predictions by contextual importance and utility,” in *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, Springer, 2019, pp. 95–109.

[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[48] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, *et al.*, “Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, 2020.

[49] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, “Explaining anomalies detected by autoencoders using shap,” *arXiv preprint arXiv:1903.02407*, 2019.

[50] S. Knapič, A. Malhi, R. Saluja, and K. Främling, “Explainable artificial intelligence for human decision support system in the medical domain,” *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 740–770, 2021.

[51] D. M. Powers, “Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation,” *arXiv preprint arXiv:2010.16061*, 2020.

Appendices

Appendix 1 - Non-exclusive licence for reproduction and publication of a graduation thesis

I Molika Meas

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "XAI-based Fault Detection, Diagnosis and Monitoring Method for Air Handling Units", supervised by Juri Belikov
   (a) to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   (b) to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the nonexclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

Appendix 2 - Expert Survey

Explainable Fault Detection and Diagnosis Methods for Air Handling Units

Research Survey

Goal: The goal of the survey is to assess the explanations' influence on the decision-making for fault diagnosis tasks and to compare various visualization techniques. The focus is on the delivery of the explanations.

Domain Requirements

This is a sample of an auto generated explanation for AHU fault, where the variables highlighted in red and blue are variables that have positive and negative influence on the fault likelihood, respectively.

Time: "03-25 15:45"
Predicted Fault: Heating coil valve leak
Fault Probability: 1.00

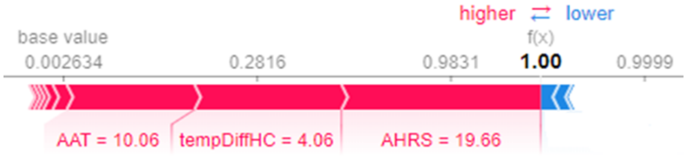


Figure 23. (Heating coil valve leakage) standard SHAP plot for individual explanations.

Based on the above plot, please **rate**, on a scale of 1 to 10, how important these criteria are in helping to understand whether a fault explanation actually represents a fault.

CR1: I want to be able to choose and view more variables (i.e., AHCVO) in addition to the auto generated features as shown in the plot above (AAT, tempDiffHC, AHRS). ______ /10

CR2: I want to be able to visualize the short history of fault to understand what has happened prior to the fault. ______ /10

CR3: I want to be able to view only the most important variables that influence the likelihood of the fault ______ /10

CR4: I want to view in terms of probability how much each feature affects the fault likelihood. ______ /10

Other important criteria you would like to add: ____

Visualization design

The following are the sample explanations for AHU faults visualized in 3 different ways to explain each type of fault. Please indicate if the explanation is convincing and rate on the scale of 1 to 10 how is your satisfaction with each type of visualization.

* All the samples shown here are observations only when AHU is switched on.

Predicted Fault: Fan Pressure Sensor Malfunction

(a).

Time: "03-11 16:00"

Fault Probability: 1.00

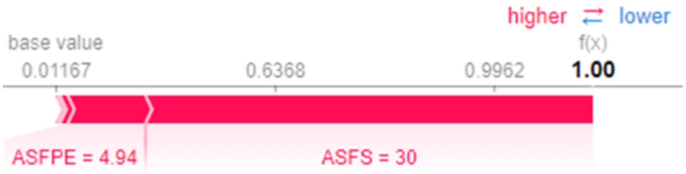


Figure 24. (Fan pressure sensor malfunction) SHAP explanation for fault at a specific timestamp.

Does the above plot explain fan pressure sensor malfunction? (Strongly Agree / Somewhat Agree / Neutral / Somewhat Disagree / Strongly Disagree) ________

Your satisfaction with the graphical representation: ______/10

(b).

Time: "03-11 19:45"

Fault Probability: 0.99

Does the above plot explain fan pressure sensor malfunction? (Strongly Agree / Somewhat

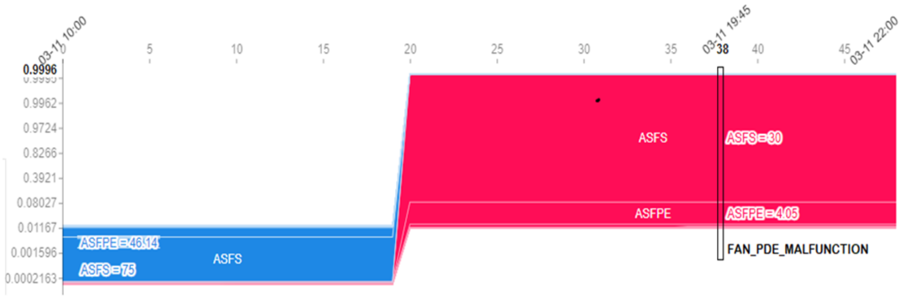


Figure 25. ((Fan pressure sensor malfunction) SHAP plot for sliding window explanation visualized from “03-11 10:00” to “03-11 22:00”; x-axis represents 15min time instances.

Agree / Neutral / Somewhat Disagree / Strongly Disagree) ________

Your satisfaction with the graphical representation: _____/10

(c).

Time: “03-11 18:00”

Fault Probability: 97.73%

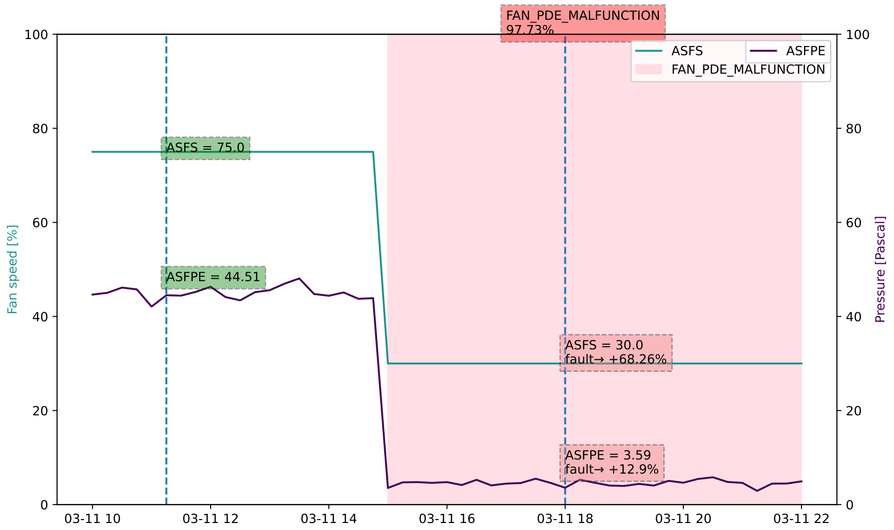


Figure 26. (Fan pressure sensor malfunction) Modified SHAP plot for sliding window explanation; x-axis represents 15min time instances.

Does the above plot explain fan pressure sensor malfunction? (Strongly Agree / Somewhat Agree / Neutral / Somewhat Disagree / Strongly Disagree) ________

Your satisfaction with the graphical representation: _____/10

Predicted Fault: Heating coil valve leakage

(a).

Time: “03-25 15:45:00”

Fault Probability: 1.00

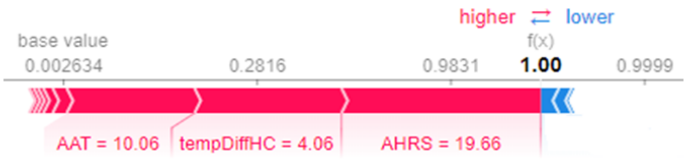


Figure 27. (Heating coil valve leakage) standard SHAP plot for individual explanations.

Does the above plot explain fan pressure sensor malfunction? (Strongly Agree / Somewhat Agree / Neutral / Somewhat Disagree / Strongly Disagree) ________

Your satisfaction with the graphical representation: _____/10

(b).

Time: “03-25 15:00”

Fault Probability: 0.99

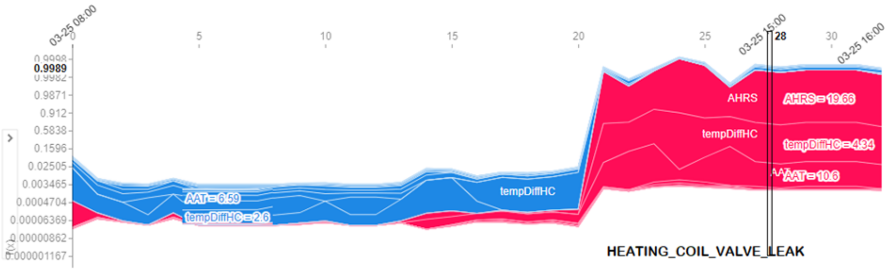


Figure 28. (Heating coil valve leakage) SHAP plot for sliding window explanation, visualized from “03-25 08:00” to “03-25 16:00”.

Does the above plot explain fan pressure sensor malfunction? (Strongly Agree / Somewhat Agree / Neutral / Somewhat Disagree / Strongly Disagree) ________

Your satisfaction with the graphical representation: _____/10

(c).

Time: “03-25 13:45”

Fault Probability: 97.53%

Does the above plot explain fan pressure sensor malfunction? (Strongly Agree / Somewhat Agree / Neutral / Somewhat Disagree / Strongly Disagree) ________

Your satisfaction with the graphical representation: _____/10

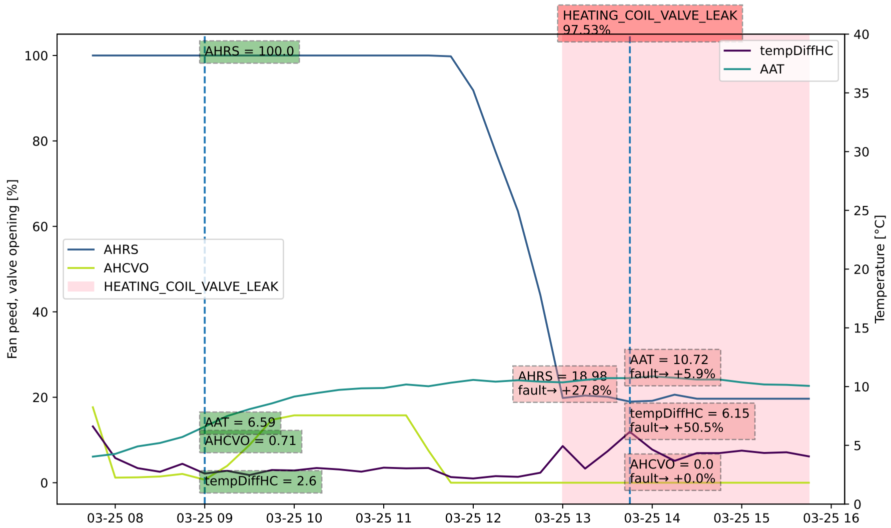


Figure 29. (Heating coil valve leakage) Modified SHAP plot for sliding window explanation.