



POLITECNICO DI MILANO
Department of Energy
DOCTORAL PROGRAMME IN Energy and Nuclear Science and Technology

Data driven Fault Detection and Diagnostics for HVAC systems in buildings

Doctoral Dissertation of:
Mohammad Abdollah Fadel Abdollah

Supervisor:
Prof. Rossano Scoccia

Co-Supervisor:
Prof. Marcello Aprile

Tutor:
Prof. Livio Mazzarella

The Chair of the Doctoral Program:
Prof. Vincenzo Dossena

2024 – XXXVI Cycle

Abstract

Building systems, particularly Heating, Ventilation, and Air Conditioning (HVAC) systems, are prone to various operational issues that can significantly impact their performance. These issues, ranging from sensor malfunctions to equipment failures and suboptimal system operations, often result in a cascade of negative consequences. These include excessive energy consumption, inflated maintenance expenses, compromised indoor environmental quality. Research indicates that such system faults and inefficient controls can lead to energy wastage of 15% to 30% in buildings. In recent years, the widespread implementation of building automation systems, coupled with advancements in data analytics, sensor technology, and machine learning algorithms, has sparked growing interest in data-driven Fault Detection and Diagnostics (FDD) for HVAC systems. Many studies have explored traditional machine learning and deep learning models trained in supervised, semi-supervised and unsupervised ways. However, several gaps in the literature persist. These include a lack of studies in residential contexts, unclear understanding of temporal dependencies' importance, insufficient research on addressing labelled data challenges for unseen fault detection, and limited studies using real building data. This thesis aims to address several objectives. Firstly, it investigates the efficacy of select supervised methods in residential settings, focusing on a minimalist feature set to optimize practicality and efficiency. Secondly, the research evaluates the potential of multivariate time series classification algorithms for FDD, exploring their capacity to capture temporal patterns in HVAC system behaviour. Lastly, the study develops and tests a novel self-supervised learning algorithm. This approach leverages unlabelled data to accelerate the data annotation process, a step in overcoming the scarcity of labelled datasets in real-world scenarios. The performance of this self-supervised method is rigorously assessed using data from an actual building. To address the first objective, a case study was done on a residential building using simulated data. Extreme Gradient Boosting exhibited an accuracy of 85% using minimal features while accounting for the risk of overfitting. Moreover, interpretability methods were used to insure model transparency both in global and local model's decision. For the second objective a benchmarking study was done on multiple multivariate time series classification algorithms used against open-source datasets. Deep learning-based algorithms exhibited the highest performance with an F1 score of 0.92, 0.85 and 0.61 respectively on the three

datasets used. For the third objective, an innovative transformer-based self-supervised method was developed to leverage unlabelled data for fault detection. The method was coupled with dynamic thresholding technique called peak of threshold to detect more subtle faults. This approach aims to address the critical challenge of data annotation in real-world applications. The method was tested using data from a university campus building. The algorithm successfully identified various faults, primarily in the monitoring system, but also uncovered issues in the air handling unit scheduling.

Summary

Table of Contents

List of Figures	9
List of Tables	12
Nomenclature	13
1 Introduction	17
1.1 Motivation	17
1.2 Literature review	19
1.2.1 Overview of the literature	19
1.2.2 Data cleaning, preprocessing and feature engineering	20
1.2.3 Supervised algorithms for FDD	21
1.2.4 Unsupervised algorithms for FDD	24
1.2.5 Fault types	26
1.3 Objectives	26
1.4 Work structure	29
2 Data driven FDD for hydronic and monitoring systems in a residential building	30
2.1 Challenges of FDD in the residential sector	30
2.2 Case study	30
2.2.1 Envelope properties and setpoints	31
2.2.2 Mechanical system scheme	33

2.2.3	Dynamic simulations and faults implemented.....	34
2.3	Automated fault detection and diagnostics.....	36
2.3.1	Data driven models.....	36
2.3.2	Data preprocessing.....	36
2.4	Results and discussion.....	38
2.4.1	Performance on evaluation metrics.....	39
2.4.2	Model behaviour interpretation.....	43
3	Multivariate time series classification algorithms benchmarking.....	51
3.1	Background.....	51
3.1.1	Time series definition and status in FDD research.....	51
3.1.2	Distance-based classifiers.....	52
3.1.3	Interval based classifiers.....	53
3.1.4	Convolutional based classifiers.....	54
3.1.5	Deep learning-based classifiers.....	54
3.1.6	Dictionary based classifiers.....	55
3.2	Datasets description and preparation.....	56
3.2.1	Datasets description.....	57
3.2.2	Datasets preparation.....	66
3.3	Results.....	67
3.3.1	Performance evaluation.....	67
3.3.2	Runtime analysis.....	73
3.3.3	Comparing performance across classifiers categories.....	75
3.3.4	Comparison with tree-based ensemble method.....	77
4	Self-Supervised Transformer based architecture for fault detection.....	79
4.1	Background.....	79

4.2	Transformer introduction & applications.....	81
4.2.1	Vanilla transformer.....	81
4.2.2	Transformers for time series and anomaly detection	83
4.3	Methodology of fault detection using a Transformer architecture.....	84
4.3.1	Core Architecture	84
4.3.2	Self-supervised learning pre-training.....	88
4.3.3	Dynamic thresholding and fine tuning.....	90
4.4	Case study	92
4.4.1	Building envelope and HVAC system	92
4.4.2	Data description	95
4.5	Results and discussion	97
5	Conclusions	104
	Bibliography	110
	Appendix.....	129

List of Figures

Figure 1: Illustration of proposed FDD solution.....	28
Figure 2: Apartment plan, areas of the rooms and adjacency information.	31
Figure 3: Weekly occupancy schedule.....	33
Figure 4: Schematic of the plant design [Source: Ariston].	34
Figure 5: Simplified system scheme with the faults implemented highlighted with red dashed line.	35
Figure 6: Representation of cyclical temporal features.	38
Figure 7: Learning curve Random Forest.	40
Figure 8: Learning curve XGboost.	40
Figure 9 Learning curves of fine tuned Random Forest.	41
Figure 10 Learning curves for the fine-tuned XGboost.....	42
Figure 11: confusion metrics of XGboost for both training and testing sets.	43
Figure 12: part of the information distillation from XGboost. Full figure can be found here.....	44
Figure 13: An example of a local prediction path. Full SVG figure can be found here.	45
Figure 14 Global interpretation of the model using aggregated SHAP value.	47
Figure 15: Features importance for predicting Class 8 (constant noise on living room sensor). .	48
Figure 16 Local explanation of one prediction of Class 8 using SHAP.....	49
Figure 17: Local prediction probabilities. The actual class is Class 3 (highlighted), and the model predicted it.	50
Figure 18: Local prediction probabilities. The actual class is Class 0 (highlighted), and the model predicted Class 3.	50
Figure 19: Schematic of the boiler plant taken from [106].....	58
Figure 20 Schematic of the chiller plant taken from [106].....	60
Figure 21: Schematic of the single duct AHU system taken from [106].	64
Figure 22: Expanding window for data splitting.	67
Figure 23: Heat map of F1 scores of all algorithms across datasets.	68
Figure 24: Radar plot for each algorithm performance on all three datasets.....	72
Figure 25: Runtime comparison across datasets.	73

Figure 26: Heat map of runtimes across datasets.....	74
Figure 27: Box plot of the average algorithm's performance across datasets by category.	76
Figure 28: Average accuracy vs runtime per classifier category.....	76
Figure 29: on the left figure (starting from the bottom): the main architecture of the algorithm the steps of preprocessing and encoding of the data to the model dimension (d) are displayed; on the right the modelling and reconstruction of the data to the origin.....	86
Figure 30 Pre-training step overview.....	90
Figure 31: a) A picture of the building after renovation and localization at Politecnico di Milano, Bovisa Campus; b) plan and a section of the building.....	93
Figure 32 Schematic of the HVAC system and the positions of the sensors of the monitoring system.	94
Figure 33: actual system implementation. a): top: Air handling unit; bottom: radiant floor; b): indoor unit of the heat pump.....	94
Figure 34 Correlation between the features.....	97
Figure 35: Demonstration of the fault detection process in case of both sequential and point anomalies.	98
Figure 36 Indoor temperature sensor readings, the reconstructed time series from the algorithm and the anomaly scores as a demonstration of sequential anomaly in the readings.	100
Figure 37 Summary of the outcome for all the features. The highlighted areas are the periods labelled as faults.....	101
Figure 38: Air handling unit power consumption. Scheduling fault detected is highlighted in red.	102
Figure 39 Original time series vs reconstructed time series vs reconstructed for AHU power consumption with average attention scores for every time step.	103
Figure 40: Dimensionality reduction of SD-AHU dataset using PCA.	129
Figure 41 Confusion matrix of LSTMFCNC.	130
Figure 42 Time series plot of AHU cooling coil valve control signal for the four intensities of the cooling coil valve leakage.....	131
Figure 43 Histogram of the data distribution of AHU cooling coil valve control signal for the four intensities of the cooling coil valve leakage.	131

Figure 44 Time series plot of zone4 air temperature for the four intensities of the cooling coil valve leakage. 132

Figure 45 Histogram of zone4 air temperature for the four intensities of the cooling coil valve leakage. 132

List of Tables

Table 1: Case study characteristics.	31
Table 2: Infiltration and ventilation rates for Summer and Winter.	32
Table 3 Faults implemented in the model.	35
Table 4: Results of the algorithms used.	39
Table 5 Input scenarios and fault imposed in the boiler plant model. Taken from [106]	58
Table 6 Data points of the boiler datasets used in training	59
Table 7: Input scenarios and fault imposed in the chiller plant model. Taken from [106].	59
Table 8 Data points of the Chiller plant datasets used in training	61
Table 9: Input scenarios and fault imposed in single duct AHU model. Taken from [106].	64
Table 10 Data points of the Single duct AHU datasets used in training	65
Table 11 Results for the three datasets.	72
Table 12: Performance comparison between XGBoost and LSTMFCNC across datasets	77
Table 13: Model architecture breakdown.	86
Table 14 The measurements from the system that is used in the training of the model.	95
Table 15 Measurement uncertainty for used features.	96

Nomenclature

AFDD Automated Fault Detection and Diagnosis

AI Artificial Intelligence

ACODAT Autonomous Cycle of Data Analysis

AHU Air Handling Unit

ARM Association Rule Mining

ARX Autoregressive with Exogenous Input

BAS Building Automation System

BMS Building Management System

BPNN Back Propagation Neural Network

BRT Boosted Regression Trees

CART Classification and Regression Trees

CC-RF Classifier Chains with Random Forest

CNN Convolutional Neural Network

DMG Diagnostic Multi-query Graphs

DT Decision Tree

ECLAT Equivalence Class Transformation

FCU Fan Coil Unit

FDD Fault Detection and Diagnosis

FP-growth Frequent Pattern Growth

GAN Generative Adversarial Network

GMM Gaussian Mixture Model

HMM Hidden Markov Model

HVAC Heating, Ventilation and Air Conditioning

IGFF Information Greedy Feature Filter

KNN K-Nearest Neighbour

LSTM Long Short-Term Memory

MLP Multi-Layer Perceptron

MSPCA Multi-Scale Principal Component Analysis

PAM Partitioning Around Medoids

RF Random Forest

RLT Run Length Transform

RNN Recurrent Neural Network

SAE Supervised Auto-Encoder

SFA Slow Feature Analysis

SVM Support Vector Machine

TARM Temporal Association Rule Mining

XGBoost Extreme Gradient Boosting

1D-CNN One-Dimensional Convolutional Neural Network

AE Autoencoder

CIF Canonical Interval Forest

DTW Dynamic Time Warping

FCN Fully Convolutional Network

GAP Global Average Pooling

GRU Gated Recurrent Unit

LSTMFCN Long Short-Term Memory Fully Convolutional Network

MLCN Multivariate Long Short-Term Memory Fully Convolutional Network

MTSC Multivariate Time Series Classification

MUSE Multivariate WEASEL

PCA Principal Component Analysis

PPV Positive Predictive Value

ResNet Residual Network

ROCKET Random Convolutional Kernel Transform

SD-AHU Single-Duct Air Handling Unit

SFA Symbolic Fourier Approximation

TSC Time Series Classification

TSF Time Series Forest

VAV Variable Air Volume

WEASEL Word Extraction for time Series Classification

CV Computer Vision

GAN Generative Adversarial Network

GPD Generalized Pareto Distribution

HP Heat Pump

LSTM Long Short-Term Memory

MLE Maximum Likelihood Estimation

NLP Natural Language Processing

POT Peak Over Threshold

QKV Query-Key-Value

ReLU Rectified Linear Unit

SSL Self-Supervised Learning

SVG Scalable Vector Graphics

t2v Time2Vec

VAE Variational Autoencoder

1 Introduction

This section is divided into four subsections. First the motivation of this thesis will be presented. Secondly, an overview of the literature is discussed. Thirdly the objectives and research questions this thesis trying to answer are presented and lastly the work structure of the upcoming sections.

1.1 Motivation

Buildings are complex integrated systems consisting of multiple components and subsystems and sensors. According to the United Nations Environment Program, 39% of the carbon dioxide emissions is attributed to the building systems [1]. Due to malfunctioning in the control, monitoring systems and equipment, 30% of the energy use in buildings is wasted [2], [3]. It is estimated that 0.37 to 17.96 EJ of additional energy consumption is due to faulty operations in buildings systems in the US [4]. Automated Fault Detection and Diagnosis (AFDD) provide a solution to tackle this problem [4],[5]. Many AFDD methods have been developed in both component level and whole building level. Kim, Katipamula and Brambly provided a comprehensive classification and review of system AFDD for Heating, Ventilation and Air conditioning (HVAC) [6], [7], [8]. According to this classification, AFDD methods can be divided into two main categories: qualitative and quantitative model-based methods (rule-based and physics-model based), and process history-based methods (mostly data driven and machine learning-based methods). Qualitative and quantitative methods are very popular among engineers and researcher as they are clear and easy to understand, however, due to their low scalability, their specificity and high development cost, the market adaptation has been low [9]. Therefore, process history-based AFDD methods received greater attention in the past decade due to their scalability

and low implementation cost. However, process history-based performance depends heavily on the quality of the data used for training [10]. The majority of the AFDD methods are developed and evaluated using simulated system data [11], [12]. This is because implementing faults, cleaning and handling real buildings data are challenging. Gradual faults have an impact overtime that is very challenging to detect [13]. There are also the issues with the data quality coming from the Building Automation System (BAS), like: noisy sensors, missing data, sensor faults and sensors accuracy). Many studies attempted to detect and diagnose anomalies in buildings by analysing the daily energy patterns[14], [15].

Many challenges face any FDD (Fault Detection and Diagnosis) system in all types of buildings [16]. These challenges are multifaceted and include: the scarcity of labelled data for abnormal or faulty operations [17], which hinders the development of accurate fault detection models; issues of scalability, as solutions need to be applicable across diverse building systems and sizes; transferability concerns, where models trained on one building may not perform well on another; and the critical need for interpretability of results [16], [18], [19]. Traditional approaches to HVAC fault detection and diagnostics have relied heavily on manual inspections, rule-based systems, and simplistic threshold alerts. However, these methods are often inadequate in addressing the complexity of modern HVAC systems, which integrate numerous components and operate under varying conditions. The advent of data-driven techniques, powered by machine learning and artificial intelligence, offers a promising solution to these challenges. These advanced methods have the potential to detect subtle anomalies, predict impending failures, and provide deeper insights into system behaviour. However, the application of data-driven FDD in real-world HVAC systems, particularly in the residential sector, remains limited. This gap between research advancements and practical implementation motivates the need for comprehensive studies that address the unique challenges of applying data-driven FDD to HVAC systems, including issues of data quality, model interpretability, and adaptability to diverse building types. A key motivating factor for this research is the scarcity of labelled fault data in HVAC systems, which poses a significant barrier to the development and deployment of supervised learning approaches. Unlike other domains such as computer vision or natural language processing, where large datasets of labelled examples are often available, HVAC fault data is typically sparse and challenging to obtain. This scarcity is due to the infrequent occurrence of faults, the cost and disruption associated with inducing faults for data collection, and the need for expert knowledge to accurately label fault

data. Another factor is the lack of studies of data driven FDD in the residential context. This is usually due to the limited amount of data available in that context plus the relatively lower economic incentive compared to the commercial one.

1.2 Literature review

In this section, the literature review will be presented. The section is divided into subsections in function of the steps used in the process of FDD in HVAC systems I introduce in Figure 1.

1.2.1 Overview of the literature

Fault detection and diagnostics techniques for HVAC systems are typically divided into two main categories: knowledge-driven and data-driven approaches [20]. Knowledge-driven techniques are based on physical principles and are further divided into model-based and rule-based approaches. Model-based methods use physical or statistical models as benchmarks, identifying faults when observed values significantly differ from the model's predictions [21], [22]. Rule-based approaches, on the other hand, rely on diagnostic rules developed by experts in the field [23], [24]. These knowledge-driven methods offer transparent reasoning processes, but developing precise models or extensive rule sets is complex and time-intensive, given the diversity and intricacy of HVAC systems. In contrast, data-driven methods analyse operational data to automatically identify relationships between faults and associated variables, making them more practical for real-world implementation. This classification scheme, despite its limitations, was sufficient when most methods relied on physical models and was widely adopted by other reviewers, including Chen et al. [25]. However, the rapid advancement of data science has led to a shift in focus towards data-driven Fault Detection and Diagnosis (FDD) in recent review articles. Zhao et al. [26] reviewed AI-based FDD methods for building energy systems, proposing separate classification schemes for fault detection and diagnosis. Both schemes categorized studies as either data-driven or knowledge-driven. Fault detection methods were further subdivided, with data-driven approaches including classification-based, unsupervised learning-based, and regression-based methods, while knowledge-driven approaches encompassed model-based and rule-based methods. For fault diagnosis, data-driven methods included classification-based and unsupervised learning-based approaches, while knowledge-driven methods comprised inference-based (e.g., Bayesian network,

fuzzy logic) and diagnostic rule-based methods. Some methods were additionally labelled as AI-based. Mirnaghi and Haghghat [27] reviewed FDD in large-scale HVAC systems using data-driven methods, classifying FDD into model-based, data-driven, or knowledge-based categories. They further divided data-driven methods into qualitative (expert systems, fuzzy logic, pattern recognition, frequency analysis) and quantitative (statistical methods, neural networks) approaches. They defined knowledge-based methods as a combination of qualitative model-based methods (structural graphs, fault trees, qualitative physics) and data-driven subcategories (fuzzy logic, expert systems). In this section, a review of the literature for each component of the proposed solution will be presented.

1.2.2 Data cleaning, preprocessing and feature engineering

Datasets from Building Automation Systems (BAS) often suffer from incompleteness due to various issues, leading to information loss [28]. This significantly impacts Fault Detection and Diagnosis (FDD) outcomes. Literature on missing data imputation for building data covers univariate time series using statistical [29], [30] and machine learning methods [31], [32]. Recent ensemble methods have shown improved performance [33], [34]. While most imputation techniques focus on general applications, some studies address missing data specifically for FDD. Li et al. [35] proposed a filtering technique using temporal and spatial information, while Wang et al. [36] employed the expectation-maximization algorithm, both demonstrating improved FDD performance with data imputation.

Before training FDD algorithms, data preprocessing is essential to enhance performance. This step includes tasks like feature selection, data scaling, and partitioning [37]. Typically, preprocessing is done offline by selecting relevant features from historical data, which are then applied to snapshot data. In BAS, where numerous sensors may provide redundant measurements, selecting informative features is crucial for effective data-driven FDD methods [38]. Using all available features could lead to overfitting, increased complexity, and higher computational costs [39]. Therefore, feature selection is necessary to simplify the model and reduce overfitting.

Various feature selection methods, including filter, wrapper, embedded, and hybrid approaches, are used in FDD applications [40]. For example, Li et al. [41] introduced the Information Greedy Feature Filter (IGFF) to efficiently select informative features from Air Handling Unit (AHU) data,

demonstrating strong performance with the ASHRAE RP-1312 dataset [42], [43]. While filter methods are computationally efficient, they may not always yield the highest model accuracy [39]. Wrapper methods, though more computationally intensive, can optimize feature sets by evaluating different combinations, as shown by Namburu et al. [44]. However, wrapper methods are prone to overfitting [38]. Embedded methods, which combine aspects of filter and wrapper techniques within specific algorithms like Decision Trees (DT) and Random Forests (RF), have also been successfully applied [45]. Recent advancements have focused on feature extraction methods that identify "localized" patterns in time series data, aiding in feature selection. For instance, Zhang et al. [38] developed a framework integrating feature extraction and selection for whole-building FDD, which enhances model generalization by considering diverse fault behaviours. However, using wrapper methods in such contexts can significantly increase computation time.

1.2.3 Supervised algorithms for FDD

This section starts with the review of supervised models used in the literature. Later sections are dedicated to tree-based ensemble methods that are widely used in FDD applications.

1.2.3.1 Review of supervised models

Supervised learning requires labelled datasets where both input and output data are known, enabling the model to establish a mathematical function that predicts output from new inputs. This type of learning is split into classification (for discrete values) and regression (for continuous values), offering high interpretability and reliability [46]. Common supervised techniques in the HVAC field include Multi-Layer Perceptron (MLPs) [47], [48], [49], [50], and their deep learning derivatives such as Convolutional Neural Networks (CNNs) [51], [52], [53] and Recurrent Neural Networks (RNNs) [54]. MLPs are sometimes combined with other models like regression trees [55]. For example, Aguilar et al. [49] developed an Autonomous Cycle of Data Analysis (ACODAT) using Random Forests (RF) and linear regression for binary classification, followed by an MLP and RF model for behaviour prediction. Other notable methods include SVM-MSPCA for feature extraction and fault diagnosis [56], and various Bayes-based algorithms like Bayesian classifiers [57], [58], diagnostic Bayesian networks [59], [60], and Naive Bayes combined with decision trees (DTs) and RF [61]. Less common but still used algorithms in HVAC Fault Detection and Diagnosis (FDD) include XGBoost [62], [63], supervised auto-encoders (SAE) [64], diagnostic multi-query graphs (DMG) [65], hidden Markov models (HMM) [35], and ensemble

models such as those based on k-nearest neighbour (KNN) [66] or boosted regression trees (BRT) [67]. The following sub-sections are a display on the most used tree based and ensemble methods used in the literature.

1.2.3.2 Decision trees

Decision Tree (DT) models are versatile tools that can be employed for both classification and regression purposes. These models are structured using root nodes, decision nodes, and leaf nodes. The root node, also known as the parent node, represents the entire dataset and is responsible for dividing the data into two or more child nodes. When constructing a DT, key decisions involve selecting features to include as inputs, determining the criteria for splitting nodes, and deciding when to cease further branching. Notably, the growth of trees occurs randomly, necessitating the use of pruning techniques to enhance the model's performance by removing branches that rely on less important features. Pruning helps reduce the model's complexity, thereby mitigating overfitting and improving its generalization accuracy. Important concepts in DT construction include entropy and information gain, which are crucial for determining how branches are split. Entropy measures the unpredictability in a series of events and acts as an estimator, while information gain, calculated using a specific formula, provides insight into the certainty of the target variable's class. DT models (DTs) are generally straightforward to develop and offer high interpretability since they can be easily visualized and explained. Additionally, feature selection occurs implicitly within the model. However, overfitting is a common issue, as DTs can become overly complex and fail to generalize well to new data. Another potential problem is bias, which can arise if class imbalance within the dataset is not properly addressed [68].

DTs have been utilized as benchmark models to interpret MLPs [55]. In one study, DTs were analysed using a combination of temporal ARM techniques [69]. These trees were applied to detect faults during the non-transient periods of datasets, with their rule-based approach providing interpretability while the data-driven methodology enabled the automatic learning of operational patterns. However, it was necessary to reduce the granularity of the ASHRAE 1312-RP dataset to achieve optimal model performance. Additionally, DTs were used to classify data categories at the zone level, in conjunction with association methods [70]. In the context of system health monitoring, DT models were implemented due to their simplicity, enabling the differentiation between normal and abnormal operations [61]. Similar classifications were conducted in another

study, where DTs served as post-mining tools to distinguish between normal and abnormal data [71]. Moreover, DTs were employed to classify residuals generated in certain applications [72]. This model does not require complex hyperparameter optimization and outperforms other methods like Back Propagation Neural Networks (BPNN), MLP, Support Vector Machines (SVM), and LSTM networks.

1.2.3.3 Random Forest

If a Decision Tree (DT) model shows excessive variance despite applying suitable regularization techniques, it can be substituted with an ensemble of DTs known as a Random Forest (RF) model. The RF model predicts a class by averaging the outputs of multiple trees, and its accuracy tends to increase as more trees are added. This enhancement is achieved through a process called bagging, which involves sampling with replacement, along with random feature selection at each step of tree construction. This method trains ensembles of trees to achieve higher predictive accuracy. Essentially, the RF model manages to encode more complex distributions by employing highly expressive individual trees, whose variance is subsequently constrained through a voting mechanism during inference. However, training an RF model demands more computational resources compared to a DT. Additionally, RF models do not handle sparse data effectively and are not well-suited for extrapolation tasks, making classification random forests more commonly used than regression trees [46].

RF models have been studied and compared with autoregressive (ARX) models for predicting fault-free operation, specifically in predicting the total heating capacity of buildings, followed by fault detection using residual analysis [73]. While RF models are more challenging to interpret, they are less prone to overfitting and offer an efficient strategy for nonlinear modelling. Parzinger et al. [74] expanded their previous research by creating an algorithm to determine the optimal decision rule for identifying errors. Wu et al. [75] introduced an innovative hybrid method combining classifier chains with the integrated RF approach (CC-RF) to address concurrent faults in RLT units as a multi-label problem.

RF models are frequently used in hybrid approaches to boost accuracy and are particularly recommended when many classes need to be predicted. Although the lack of interpretability can be a drawback in the field of HVAC system monitoring, RF models have demonstrated strong predictive performance.

1.2.3.4 *Extreme Gradient Boosting*

Some limitations of Random Forest (RF) modelling can be addressed through the use of XGBoost, an advanced ensemble learning technique rooted in gradient boosting. This powerful algorithm operates by allowing each successive predictor to correct the cumulative errors made by its predecessors. In XGBoost, the individual models that comprise the ensemble, typically DTs, are constructed sequentially, each one focusing on the residuals of the previous models. These individual classifiers then combine to form a robust and more accurate model [76].

In [62], a hybrid reference model known as multi-region XGBoost was applied as a classifier, incorporating a type of DT called Classification and Regression Trees (CART). The results demonstrated the model's high accuracy in identifying errors, suggesting that it generalizes well and serves as a reliable and efficient model for FDD. This model outperformed both Support Vector Machines (SVM) and the standard XGBoost model. Additionally, XGBoost was used as a prediction model for energy consumption in [63], where it was combined with a novel dynamic threshold technique for FDD. This method effectively detected fault occurrences and dynamically adjusted the threshold value based on the real-time moving average and moving standard deviation of the forecasts.

Although XGBoost is not yet widely used in HVAC FDD applications, it excels in handling large datasets and class imbalances, often outperforming models like SVMs in FDD tasks. Furthermore, due to its foundation in decision tree boosting, which involves a sampling technique, XGBoost models are resistant to distribution skewness.

1.2.4 Unsupervised algorithms for FDD

Unsupervised learning is applied to discover patterns in datasets where the data are unlabelled. Unlike supervised learning, which predicts outputs, unsupervised learning focuses on identifying relationships between data instances based on shared patterns. This type of learning can be further categorized into clustering and association techniques. Clustering involves grouping data instances that exhibit similar patterns, while association uncovers the underlying rules that define the data structure.

Clustering approaches like k-means with squared Euclidean distances, Ward's linkage using Euclidean distances, and Gaussian mixture model (GMM) clustering have been employed in

studies [77], [78], [79]. Additionally, association rule mining (ARM) techniques have been implemented using methods such as the FP-growth algorithm [80], [81], episode-based association methods [70], and Apriori, ECLAT, and FP-growth algorithms [71], along with Temporal ARM (TARM) utilizing the cSpade algorithm [69]. Other specialized algorithms identified in this context include a feature extraction model known as three-way data-based kernel Slow Feature Analysis (SFA) [82], conditional Wasserstein Generative Adversarial Networks (GANs) combined with an optimized ensemble learning quality control protocol [83], and autoencoders integrated with long short-term memory (LSTM) networks [84].

A multi-regional XGBoost model based on Classification and Regression Trees (CART) was developed using a mean-shift clustering technique [62]. Clustering following the “Follow the Leader” algorithm was also performed, followed by frequency analysis to label anomalous data [55], [69]. A study presented in [77] proposed a machine learning-based multi-stage automatic fault detection system that focuses on analysing Fan Coil Unit (FCU) subsystems. This method employs sequential two-stage clustering to detect abnormal behaviour. Another clustering-based method was applied in [78], where faults were identified using Ward’s linkage method with Euclidean distances, uncovering issues that were previously overlooked by operational staff and commercial FDD tools. Furthermore, anomaly detection at the zone level was performed using various clustering techniques, such as k-means, Gaussian mixture, and agglomerative clustering algorithms, with the best algorithm selected based on the Calinski–Harabasz index [79]. Feature selection was also conducted using clustering analysis, which included agglomerative hierarchical clustering, k-means, and Partitioning Around Medoids (PAM) algorithms [71]. In a hybrid approach, a physical model was developed and evaluated through a clustering technique by measuring distances between instances to differentiate between normal operation and four types of faults [27].

The study detailed in [98] implemented a two-step rule extraction approach, utilizing both Decision Trees (DT) and Temporal Association Rule Mining (TARM). The TARM techniques were employed specifically for detecting faults during transient periods. In contrast, the research described in [70] concentrated on metadata inference without relying on semantic information. The authors developed a zone-level inference method that included classification using DTs and an association method known as episode-based association. After classifying the data points, the

association method was used to uncover functional relationships among these point classes by grouping the data through various matching strategies. In another study [71], an anomaly detection and dynamic energy performance evaluation method for HVAC systems was proposed, focusing on evaluating multiple energy performance metrics of individual buildings over short intervals, such as hourly. This method involved clustering techniques, where the ARM method was applied to each data cluster. The post-mining process was further assessed using DTs. Additionally, to automate the rule selection process and enhance the performance of the ARM method, the authors in [80] improved the post-mining of associated rules by developing a comparison-based post-mining approach. This approach involved grouping, normalizing, comparing association rules, and conducting expert rule analysis. Similarly, the post-mining process was refined in [81] by proposing the use of a fuzzy analytic hierarchy process. This method included three main criteria and six sub-criteria to evaluate the significance of each association rule. Fuzzy set theory was employed to assess the sub-criteria, and the analytic hierarchy process was used to determine the weight of each criterion and sub-criterion, resulting in a comprehensive evaluation of the rules. Finally, a k-means clustering algorithm was used to classify the rules based on their Euclidean distance.

1.2.5 Fault types

According to [85]. Faults in HVAC systems can be categorized into the three main types. Condition based faults which can be defined as the presence of an improper or undesired physical condition in a system or piece of equipment. An example of that would be a damper stuck in an AHU. Behaviour based faults which can be defined as the undesired behaviour during the operation of a component. Finally outcome based faults which is the quantifiable deviation from the correct outcome of a component.

1.3 Objectives

The research presented in this thesis aims to contribute to the foundational components of a comprehensive Fault Detection and Diagnosis (FDD) system, as illustrated in Figure 1. This proposed system architecture begins with the ingestion of both real-time (online) and historical data from a Building Management System (BMS). This data undergoes cleaning and preprocessing

to ensure quality and consistency. Historical data serves a dual purpose: it is utilized to train and evaluate a binary unsupervised classification algorithm. This algorithm is designed to distinguish between normal operating conditions and anomalous or potentially faulty states. The strength of using an unsupervised approach here lies in its ability to detect novel fault patterns without prior labelling, addressing the challenge of limited labelled fault data. Data points identified as faulty by the unsupervised model undergo manual annotation by domain experts. This crucial step enriches the system's knowledge base, with annotated faults being stored in a dedicated fault database. This database serves as a valuable resource for training subsequent supervised models, which will leverage both the annotated fault data and data from normal operations. To enhance the system's adaptability and performance across different HVAC systems, the fault database can be augmented with data from similar implementations. Furthermore, transfer learning techniques [86] can be employed to leverage knowledge gained from related systems, potentially improving the supervised model's performance on new, unseen building systems. The supervised model forms the core of the fault diagnostics capability. To address the interpretability challenge, this model should be coupled with ad-hoc interpretability methods, particularly for black-box models. These methods aim to provide insights into both global model behaviour and local decision-making processes, enhancing trust and facilitating actionable insights for building managers. Model calibration is another critical aspect, ensuring that the model outputs accurate probability estimates. This calibration is essential for reliable uncertainty quantification in each prediction, a feature that is invaluable for decision-making in building management. Various frameworks can be employed for uncertainty quantification in supervised models, including Bayesian approaches [36], [87] and conformal prediction methods [88]. The FDD system culminates in a comprehensive evaluation phase and the generation of FDD reports. These reports provide actionable insights back to the building management systems, creating a feedback loop that continually improves building performance and energy efficiency.

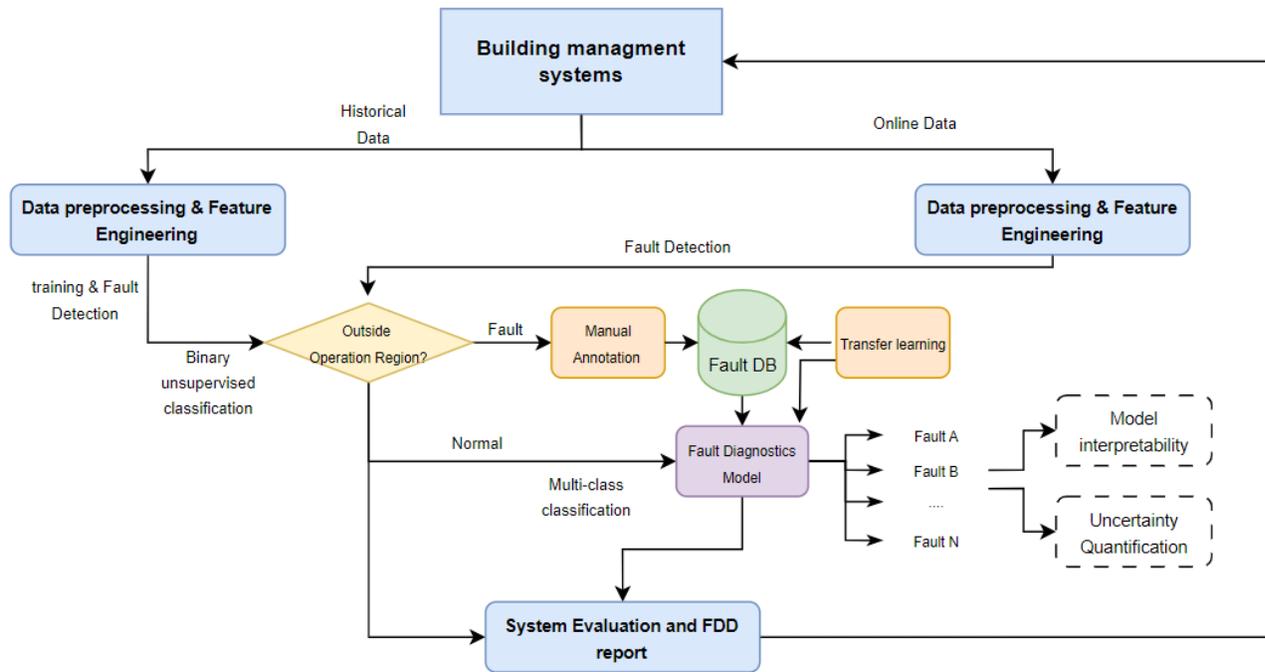


Figure 1: Illustration of proposed FDD solution.

It is important to note that this solution does not aim for full autonomy since manual annotation is needed as an intermediate step between the unsupervised and the supervised algorithms. The reasoning here is that while full autonomy should be a goal in the future but with the lack of quality data, human intervention is needed for a robust solution until sufficient data is collected and annotated for a system.

This thesis addresses three primary research questions. The first question examines the effectiveness of supervised ensemble methods in fault detection and diagnostics (FDD) within residential settings, specifically focusing on implementations that utilize a minimal feature set. The second question explores the benefits of employing multivariate time series classification algorithms, designed to inherently account for temporal dependencies in time series data, and compares their performance against other supervised learning methods. Lastly, the thesis investigates strategies to overcome the challenge of labelled data scarcity in FDD applications. It evaluates the potential of self-supervised learning approaches as a viable solution to enhance data utilization and improve model training and whether this is applicable in a real building setting. The activities addressed in the four chapters addresses the three main components in Figure 1. The first component is the data preprocessing. The work explores different methods to preprocess data

before any data modelling. The second component is the binary classification component. This is addressed in chapter four where a novel approach to detect anomalous patterns in the data. The third is the multi class classifier where the first chapters tries to answer the question about the best possible algorithm to be used and the exploration of interpretability and uncertainty quantification methods.

1.4 Work structure

The work presented in this thesis is divided into four chapters. In the introductory chapter, the work motivation and objectives are presented as well as a summarized review of the literature.

The second chapter titled “Data driven fault detection and diagnostics for hydronic and monitoring systems in a residential building” is a study on the applicability of data driven models in a residential context where a model simulating the operation of a heat pump connected to a hydronic system in a residential building is developed and faults were injected to simulate faults in hydronic and monitoring systems. A comparison between different models is presented and some global and local explanation to the models’ behaviour using ad-hoc interpretability frameworks.

The third chapter titled “Multivariate time series classification algorithms benchmarking for FDD” presents a benchmarking study of multivariate time series classification algorithms on an open-source datasets for FDD purposes to assess the usability of such models in FDD operation. Those models tend to focus on temporal features and dependencies rather than spatial dependencies that most algorithms used on tabular data use. A brief comparison of the performance of supervised models for tabular data is also presented.

The fourth chapter titled “Self-supervised Transformer based architecture for fault detection in HVAC systems.” Presents a novel self-supervised model that aims to detect faults in systems without any dependencies on label data. The architecture is an adaptation of the transformer encoder architecture coupled with a dynamic thresholding technique aimed at identifying underlying trends in multi-variate time series to flag anomalous patterns.

The closing chapter is a conclusion of all the work presented in the thesis and future research work that might be useful for future researchers.

2 Data driven FDD for hydronic and monitoring systems in a residential building

In this chapter, supervised methods will be tested in a case study of a residential context. The case study includes faults in the hydronic and monitoring systems of a residential building. The chapter is divided into three sections. First the case study will be presented including the envelope characteristics, mechanical systems and description of the simulations and faults implemented. Next section will include the description of the supervised models used in this chapter and the data preprocessing steps. Finally, the results will be presented. The goal of this chapter is to explore the applicability of data driven FDD method in a residential context, the explainability of the method used and the uncertainty quantification.

2.1 Challenges of FDD in the residential sector

The residential sector has a different constraints and limitations from the commercial sector when it comes to FDD solutions [89]. Residential buildings are often equipped with relatively less complex HVAC systems than commercial buildings. However, monitoring systems in residential buildings have limited capabilities making the buildings data poor with no standard communication protocols [90].

2.2 Case study

In this section the case study involved will be presented, including the envelope, the mechanical system used and the simulation process including the faults implemented.

2.2.1 Envelope properties and setpoints

The apartment presented consists of five rooms as shown in Figure 2. A living room with an area of 32.5 m², two bedrooms with an area of 18.8 and 20.1 m² and two bathrooms. It is adjacent to two other apartments in the North-West and North-East, while the living area and the second bedroom border on the North-West and North-East respectively with a common area that includes a stairwell and elevators. The remaining part of the perimeter walls is in contact with the external environment. Table 1 summarizes some of the features of the apartment including the U values of envelope components, net walkable area, number of occupants.

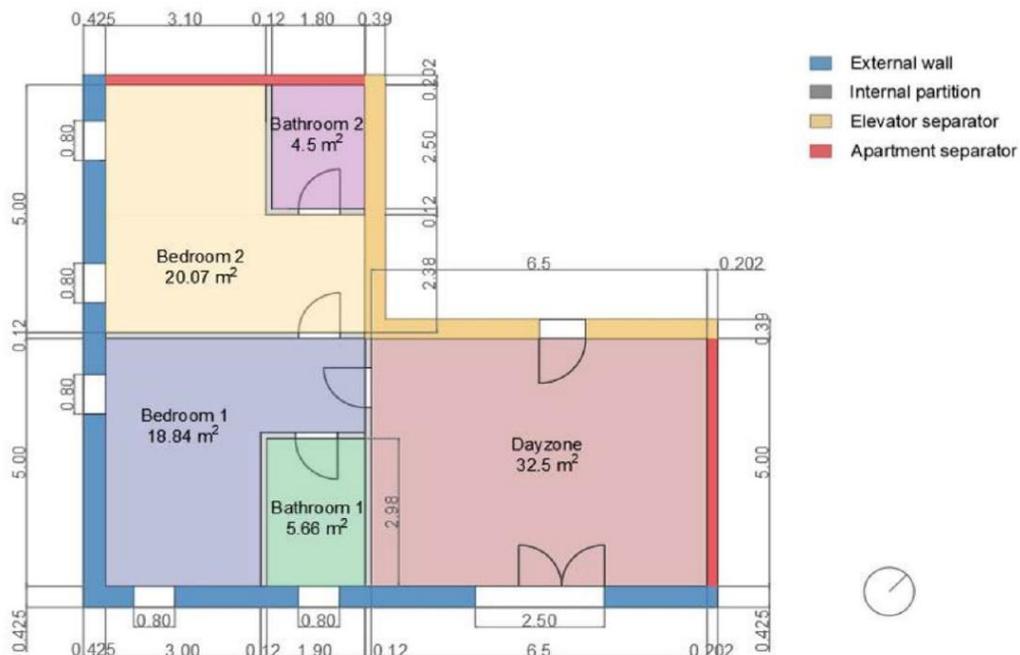


Figure 2: Apartment plan, areas of the rooms and adjacency information.

Table 1: Case study characteristics.

Net walkable surface	81.57 m ²
Local average height	2.7 m
Number of occupants	2
U values (W/m²/K)	
External walls	0.225

Internal walls towards stairwell	0.685
Partition between apartments	0.267
Windows	1.31

For the infiltration and ventilation for both living room and bedrooms the parameters were set as shown in Table 2.

Table 2: Infiltration and ventilation rates for Summer and Winter.

Parameter	Living & bedrooms
Infiltration [m ³ /s]	0.825
Daily natural ventilation (winter) & Daytime (Summer) [l/s]	4.125
Nighttime natural ventilation (Summer) [l/s]	0.825
Mechanical ventilation rate [1/h]	0.5

For the shading system, it was assigned a different degree of opening depending on the season in question. In the cooling season they were closed 85% in the living area and 70% in the sleeping area, while in the heating season they were closed at 45% in both thermal zones. The internal loads were taken as 60 W/person for the sensible loads and 20 W/person for the latent following the occupancy schedule shown in Figure 3. The set points temperatures are set to be 20 °C with a setback of 18 °C in winter and 26 °C with a setback of 28 °C in summer corresponding to the occupancy schedule.

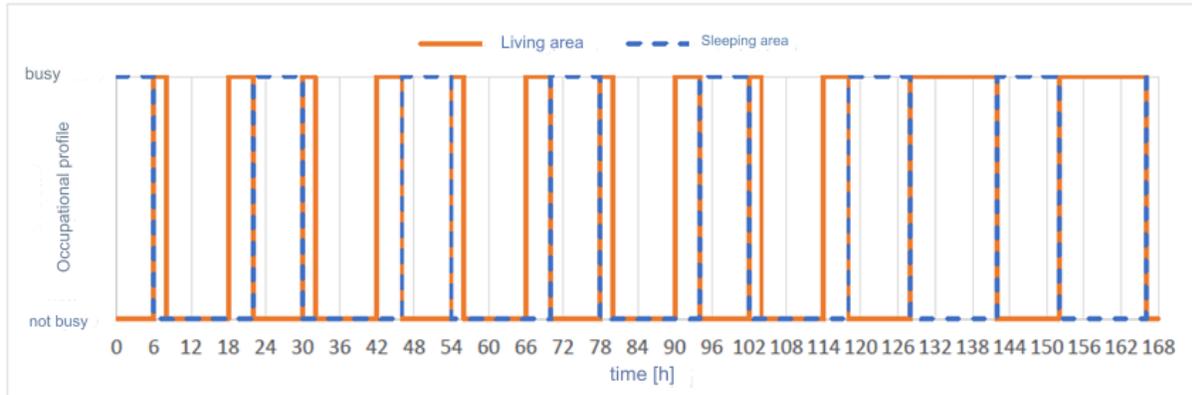


Figure 3: Weekly occupancy schedule

2.2.2 Mechanical system scheme

An entirely electric plant system was considered as shown in Figure 4. The plant is characterized by an air-water heat pump connected to a radiant floor for heating and fan coils for cooling and 180 l storage tank for the production of domestic hot water. The heat pump and the storage tank are into the perimeter structure of the house. Finally, there is also a photovoltaic system of 23.3 m² area for 3.2 kWp. The hydronic system consists of a pump regulated by a flow signal, two three-way valves that regulate the distribution of water in the system, pipes and joints that distribute it where required. This system can manage n thermal zones that can be served either by radiant floors or by fan coils, whose circuits can be operated using two-way valves.

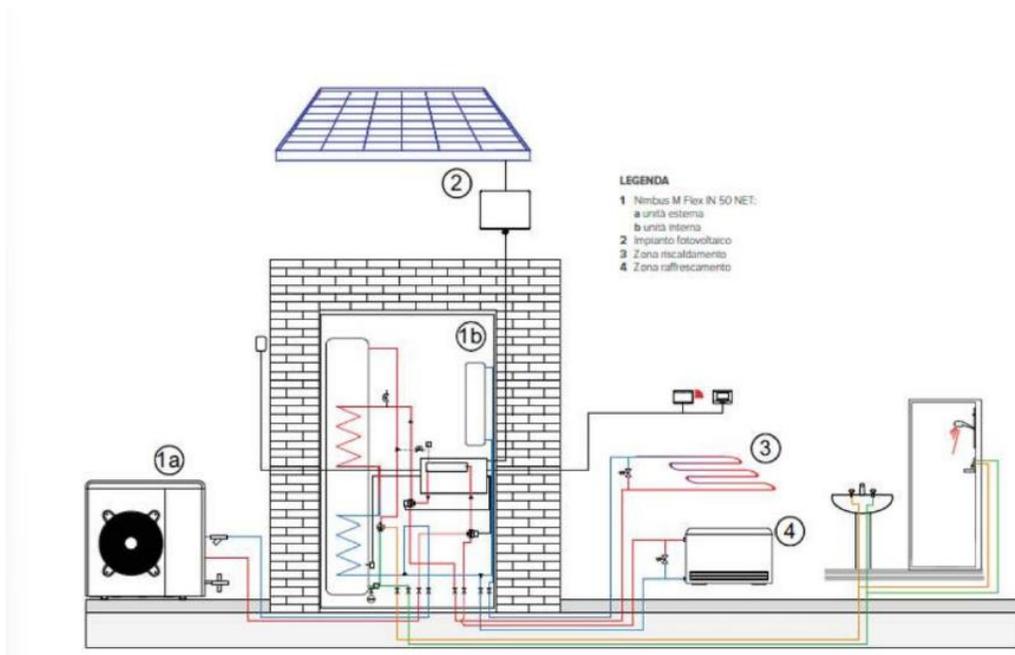


Figure 4: Schematic of the plant design [Source: Ariston].

2.2.3 Dynamic simulations and faults implemented

A detailed physical model was created for this project using Modelica language buildings and IBPSA's libraries were used to model all the components [83]. The simulations were run with a time step of 10 minutes. Simulations were run from 15th of December to the 1st of April in the winter case and from 1st of June till 1st of August for the Summer case. 12 faults were considered both in the hydronic system and the monitoring system. Figure 5 shows a simplified scheme of the system with the faults implemented highlighted. While Table 3 contains all the faults implemented and their description. The faults are simulated for the whole period in both summer and winter. The sensors faults are modelled by adding constant noise that ranges between -0.5°C to $+1.5^{\circ}\text{C}$ in the first case and a random noise with a mathematical function that adds a deviation to the measurements that adds up to $+5^{\circ}\text{C}$ to the readings. While the valve faults are added by increasing the leakage parameters provided from by the library in the component features. Finally, for the circulating pump fault, a mathematical function was provided to the input signal of the pump to decrease the required flow rate by 10%.

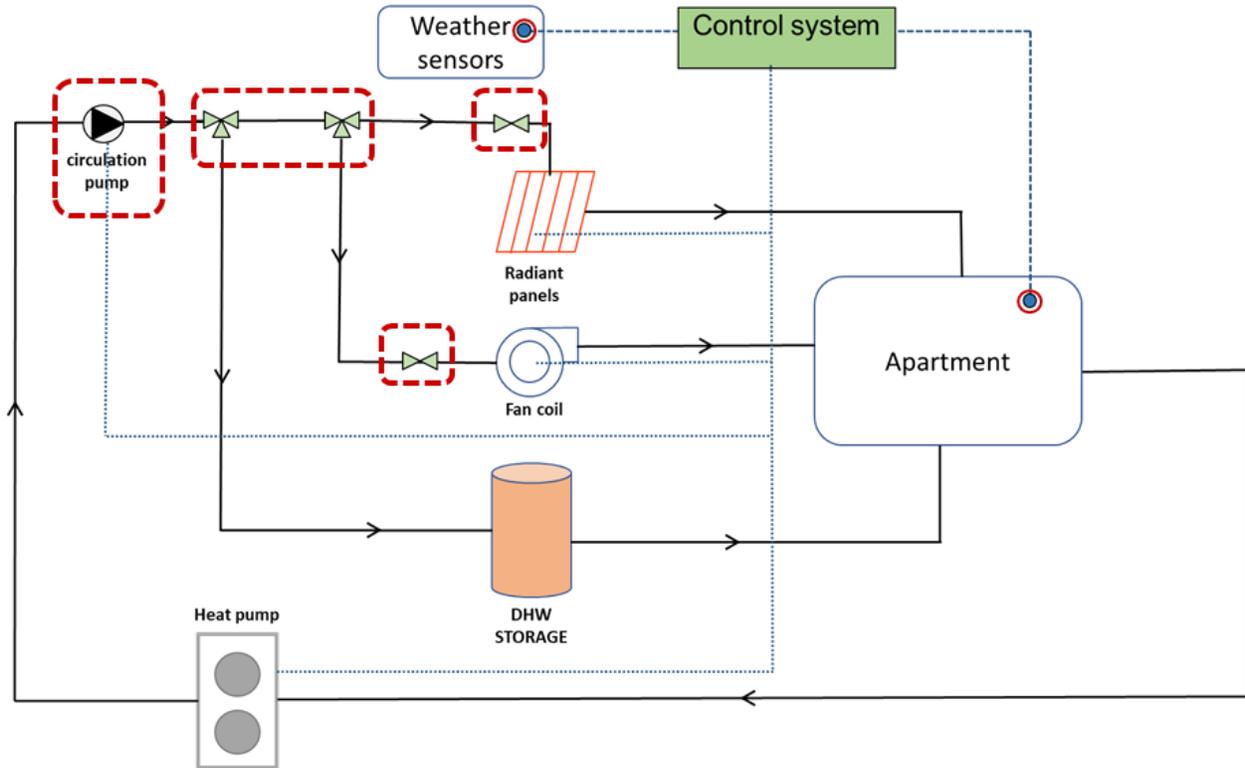


Figure 5: Simplified system scheme with the faults implemented highlighted with red dashed line.

Table 3 Faults implemented in the model.

Fault no.	Fault	Description
1 - 4	Three-way valve no.1 and 2 leakage 1% and 5%	The flow going into the three-way valve is leaking 1% and 5% of its volume to the wrong direction
5	Two-way valve no.1 blockage 1%	1% of the flow going into the two-way valve is blocked
6, 8, 10	Weather, Bedroom1 and living room dry-bulb temperature sensor with added constant noise	Random noise is added to the sensors reading. The noise is ranging from -0.5°C/+1°C
7, 9, 11	Weather, Bedroom1 and living room dry-bulb temperature sensors with increasing deviations and added noise	Deviation and random noise are added to the sensors reading. The noise is ranging from -0.5°C/+1°C, while the deviation is adding up to +5°C
12	Circulating pump inadequate flow	The circulating pump is only providing 90% of the supposed flow rate.

2.3 Automated fault detection and diagnostics

In this section, firstly, the algorithms used to detect faulty operation of the system will be introduced, then the data preprocessing step and the assessment methods used to evaluate their performance are discussed.

2.3.1 Data driven models

Data driven approach means that the decisions are made based on data analytics instead of intuition of an expert. In the fault detection and diagnostic context, statistical models are trained on the data from the system at hand from both healthy and faulty operations. The source of the data can be a simulation of the system or from experimental setup. The trained statistical model should be able to detect faulty operation of the system without the need of experts or intuition about the system.

In this section the Machine Learning (ML) algorithms used are listed. In this chapter, Random Forest (RF), K nearest neighbours (KNN), CatBoost [93], XGboost [62] and explainable boosting machine (EBM) [94] were used.

2.3.2 Data preprocessing

Data preparation is a crucial step that significantly impacts model performance. Typically, this process involves cleaning raw sensor data, handling missing values, and normalizing or scaling features to ensure all variables contribute equally to the model. Since the data in this part is a result of simulation, no cleaning or noise reduction was done. Each feature was normalized using Eq.(1).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Tree-based algorithms split the data into subsets based on certain conditions, usually involving feature values. These splits are determined by measures such as Gini impurity, information gain, or variance reduction, which are used to find the optimal conditions to partition the data. Unlike many machine learning algorithms, tree-based methods such as Decision Trees, Random Forests,

and Gradient Boosting Trees do not require feature scaling. This is due to the intrinsic properties of how these algorithms operate.

Since none of the algorithms used has an inherent representation of time, temporal features were extracted from the time using cyclical feature encoding given a feature x and period of T : Eq. (2). Where x is the original value of the cyclical feature and T is the period of the cycle (e.g., 24 for hours of the day, 7 for days of the week, 12 for months of the year). A representation of the cyclicity intended for the hours and days components is presented in Figure 6.

$$\begin{aligned}x_{\sin} &= \sin\left(\frac{2\pi x}{T}\right) \\x_{\cos} &= \cos\left(\frac{2\pi x}{T}\right)\end{aligned}\tag{ 2}$$

The features used to train the model are easy to monitor and accessible in real buildings as mentioned previously. The features used are as follows:

1. Thermal zones dry bulb temperatures (bedroom 1 and 2, bathroom 1 and 2 and the living room);
2. Inlet and outlet temperatures of the heat pump;
3. Heat pump electrical consumption.

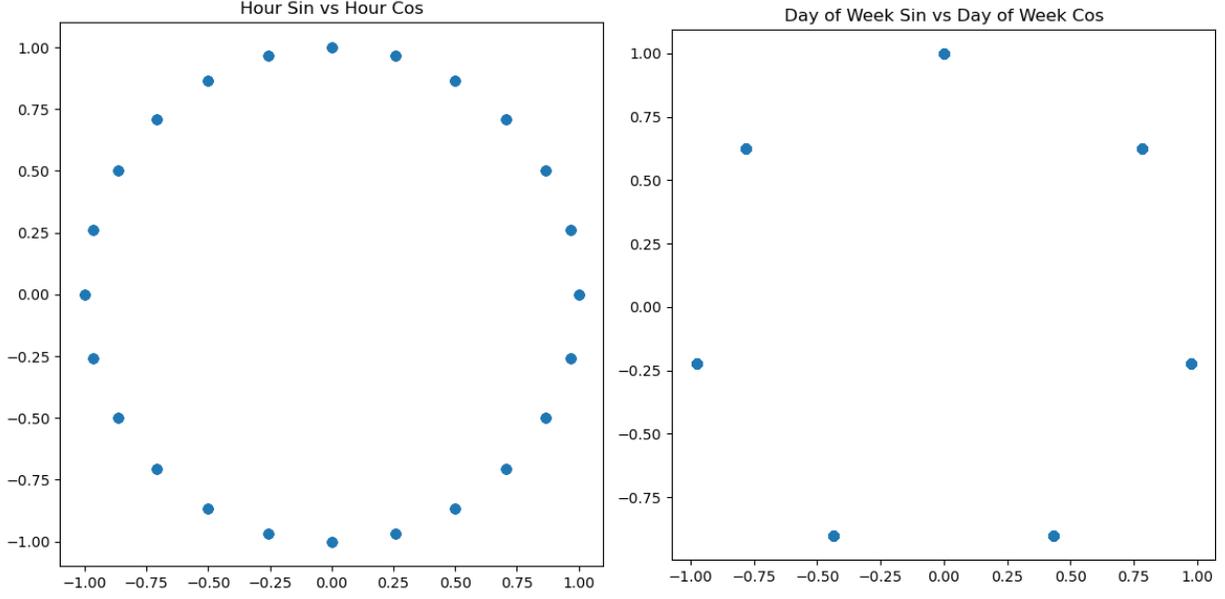


Figure 6: Representation of cyclical temporal features.

2.4 Results and discussion

In this part, Accuracy (Eq. (3)), Precision (Eq. (4)), Recall (Eq. (5)) and F1 score (Eq. (6)) were used as evaluation metrics for the algorithms. Each metric is defined in function of True or False Positives (TP, FP) and True or False Negatives (TN, FN). TP is the number of true positive samples, which indicates a positive result for both true and predicted results. FP is the number of false positive samples, which indicates a negative true label and a positive predicted result. FN is the number of false-negative samples, which indicates a positive true label and a negative predicted result. TN is the number of true negative samples, which indicates a negative true label and a negative predicted result. Accuracy calculates the ratio of the total number of correct predictions to the total number of predictions. Precision calculates the ratio of labelled faulty instances that have been correctly identified as faulty samples. Recall measures the number of faulty samples that have been correctly identified as faulty instances. F1 score balances precision and recall.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

2.4.1 Performance on evaluation metrics

Table 4: Results of the algorithms used.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Random Forest	84.0	84.7	84.0	84.3
XGBoost	83.8	83.5	83.8	83.6
CatBoost	76.4	75.5	76.4	75.8
KNN	57.1	64.9	57.1	60.1
EBM	57.2	55.5	57.2	55.4

The data was splitted 80% for training and 20% for testing. The results for the metrics are stated in Table 4. Even though Random Forest, XGboost and CatBoost seems they are performing better than the rest, Overfitting must be avoided to be able to generalize well in the field. This can be checked by plotting the learning curve for each algorithm. The learning curve shows how well the algorithm is performing under different number of training examples. The results are usually an average of cross validation with five folds.

In Figure 7 and Figure 8, the learning curves of RF and XGboost are displayed. The solid lines are the mean of the fold accuracy result while the shaded area is the standard deviation. Both algorithms showed overfitting signs especially RF where it seems the algorithm showed almost perfect score in the training set but cross validation scores mostly below 0.5. XGboost however showed less extreme sign of overfitting. The overfitting is judged by the gap between the training and testing results and how big the variation of the results among the same fold as demonstrated by the standard deviation.

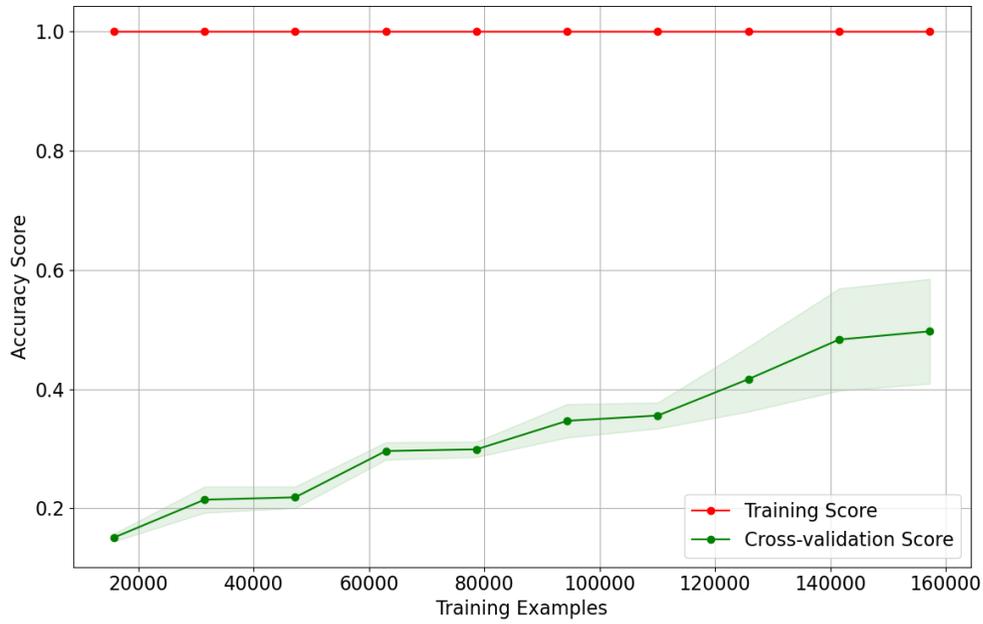


Figure 7: Learning curve Random Forest.

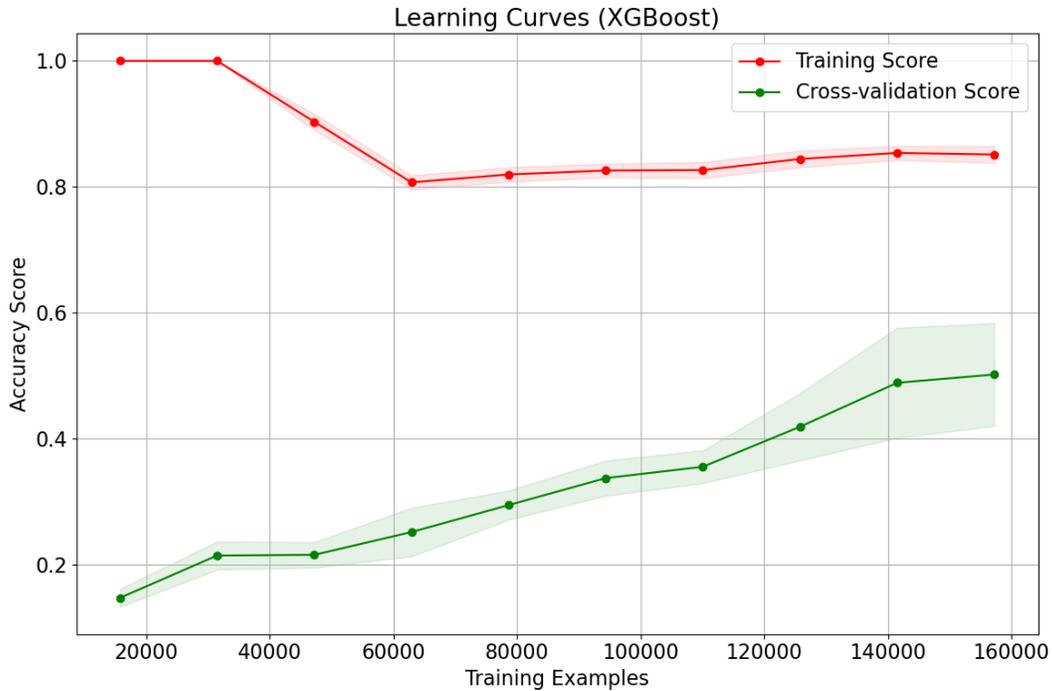


Figure 8: Learning curve XGboost.

Overfitting is usually an issue that occurs due to the over complexity of the model versus relatively simple data pattern or highly noisy data where signal to noise ratio is low causing the model to be fitted mostly on the noise causing poor generalization capability. Since we are using a minimal number of features, most likely the cause of this issue is the overcomplexity of the model. To deal

with this issue, hyperparameter tuning was performed on both algorithms using Bayesian optimization [95]. Bayesian optimization is an efficient method for hyperparameter tuning that leverages probabilistic models to optimize the objective function. Unlike traditional grid search or random search methods, which explore the hyperparameter space in a systematic or purely random manner, Bayesian optimization builds a surrogate model. In this case, a Gaussian Process, to approximate the objective function. This model predicts the performance of different sets of hyperparameters and uses this information to intelligently choose the next set of hyperparameters to evaluate. The process iteratively updates the surrogate model based on the results of each evaluation, allowing it to focus on the most promising regions of the hyperparameter space. By balancing exploration (testing hyperparameter sets in less-explored areas) and exploitation (focusing on hyperparameters that have performed well in the past), Bayesian optimization can find optimal or near-optimal hyperparameters more efficiently than exhaustive or random search methods. In Figure 9 and Figure 10 the learning curves for the fine-tuned variations are presented.

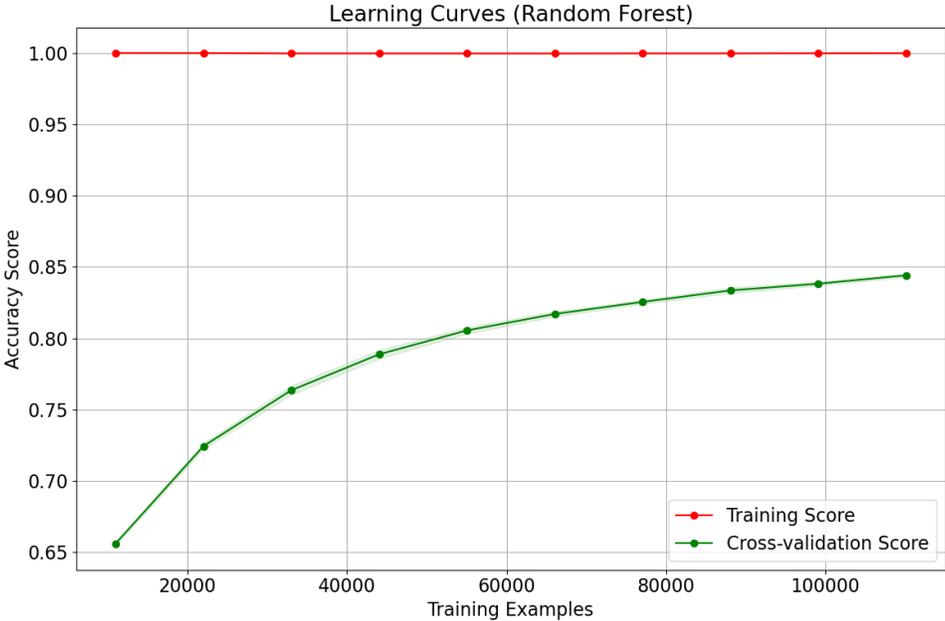


Figure 9 Learning curves of fine tuned Random Forest.

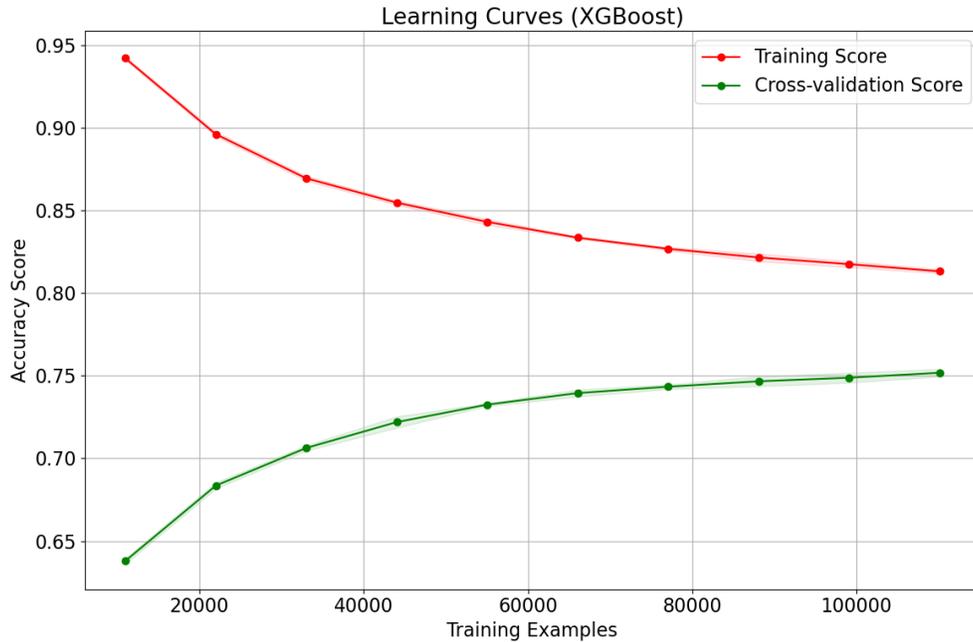


Figure 10 Learning curves for the fine-tuned XGboost.

The fine tuning was aiming to find a better parameter to achieve better accuracy without increasing model complexity. Both models showed signs of decreased overfitting. This is evident by the better conversion rate between training score and cross validation rate and the less standard deviation among the cross-validation scores. Despite the better accuracy achieved by RF, XGboost seems to achieve better conversion rate and reasonable accuracy indicating less overfitting so it was chosen as the best algorithm for further analysis.

To identify the performance of XGboost on individual classes of faults, the confusion matrix for both training and testing sets is shown in Figure 11.

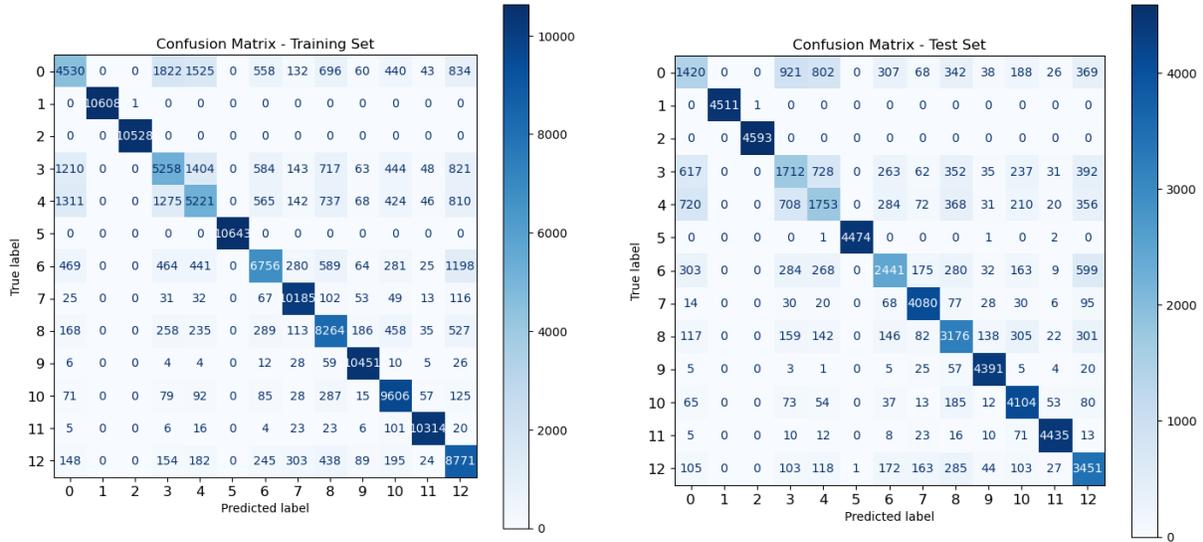


Figure 11: confusion metrics of XGboost for both training and testing sets.

The algorithm seems to struggle the most in classifying Classes 3,4 and 6 which are the leakage in the second three-way valve of 1% and 5% and the constant error in the outdoor dry bulb air temperature. Those classes are often mistakenly classified as each other or as Class 0 which is the healthy operation meaning that those faults are subtle enough to not disrupt the distribution of the data so the algorithm cannot distinguish between those faults and the healthy operation. Excluding those three faults from the faults list increase the overall accuracy of the model up to 95%. Possible solutions include generating more data for those two faults, engineer features that distinguish those two faults specifically or adding more feature to the model. Other solution is to train different models specialized on those faults and combine the results from the original model choosing the higher probability prediction. On the other hand faults like the sensors error combined with deviation and the leakage in the first three-way valve are almost perfectly classified.

2.4.2 Model behaviour interpretation

To ensure transparency and reliability, it is essential to explain and effectively communicate the behaviour of machine learning models to end users. This explainability is typically achieved through two types of interpretations: global and local. Global interpretation provides insights into the overall behaviour, performance, and key features influencing the model, while local interpretation delves into the reasoning behind individual predictions, revealing how the model arrived at specific outcomes.

In this part, two approaches will be explored to achieve both types of interpretations. The first approach will rely on distilling information from the trees directly. The model chosen has 100 estimators and each one has a depth of seven. Global interpretation in this case is difficult due to the scale of the trees but still possible if done through interactive plots. The following Scalable Vector Graphics (SVG) file is generated using the code in [96]. This file can be found through this direct link ([Trees Visualization](#)) that can be opened on any browser. Part of the visualization can be found in Figure 12. While the figure is quite challenging to depict, there are some useful insights there. The first one is that most important splits are based on the features related to the bathroom temperatures (TAirBathroom1 and TAirBathroom2), followed by the living room temperature in lower nodes. This indicated the importance of those features in the model decisions.

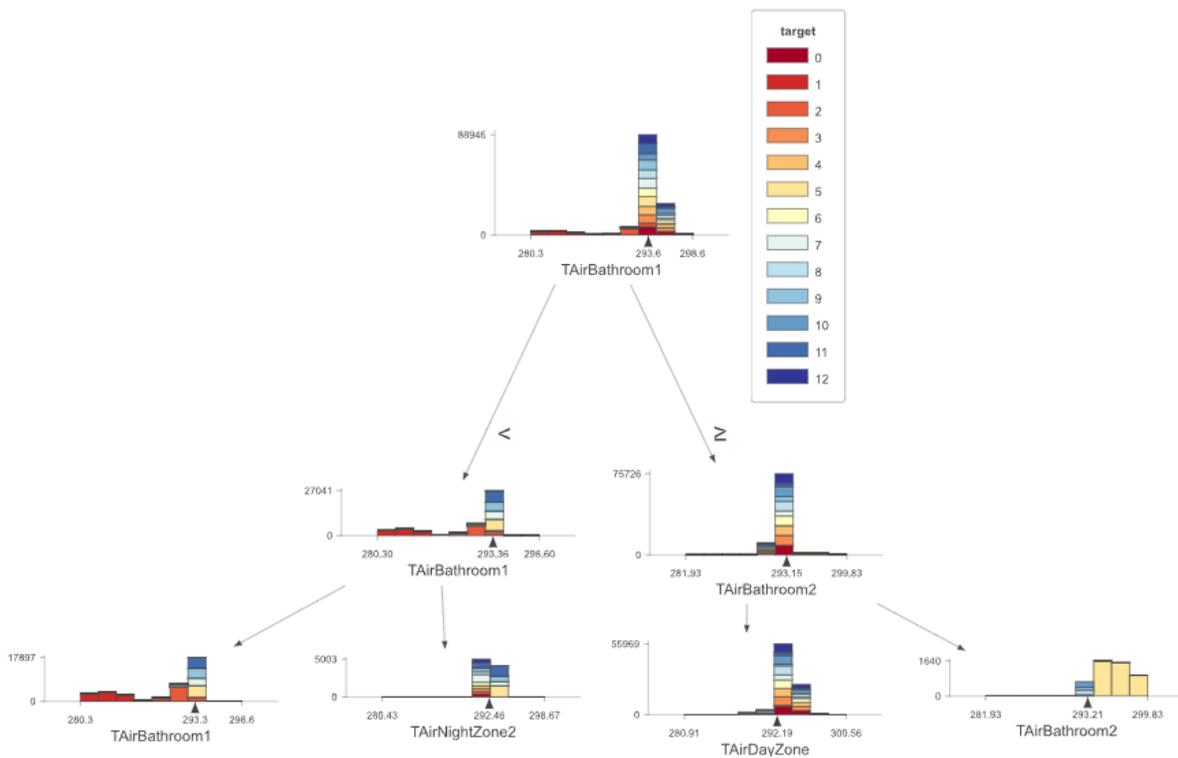


Figure 12: part of the information distillation from XGboost. Full figure can be found [here](#).

One other observation is that some classes are identified earlier in the depth of the trees and more confidently than other classes such as classes 5, 9 and 11 which are the two-way valve leakage, living room sensor fault with increasing noise and bedroom 1 sensor fault with increasing noise. As expected, most of the increasing noise faults are easily identified. On the other hand, confirming

what previously was mentioned, the faults related to the second three-way valve leakages are hardest to predict.

For local explanation, any set of points can be tracked down through the distilled tree to reach the prediction. In Figure 13, an example of a local explanation for a point prediction. It is notable that the road down the trees only used two features, and the confidence of the prediction was around 30% meaning that only 30 trees chose class 1 prediction.

Despite having full transparency in this method of interpretation, it might be challenging for non-experts to navigate. A more accessible method of interpretability can be using SHAP (SHapley Additive exPlanations) framework. introduced by Lundberg and Lee (2017) [97], SHAP offers a unified, theory-driven approach to interpreting model predictions, particularly for complex models like deep learning or ensemble models. This framework is based on concepts from cooperative game theory, specifically the Shapley value, which quantifies the contribution of each feature to the prediction of a model.

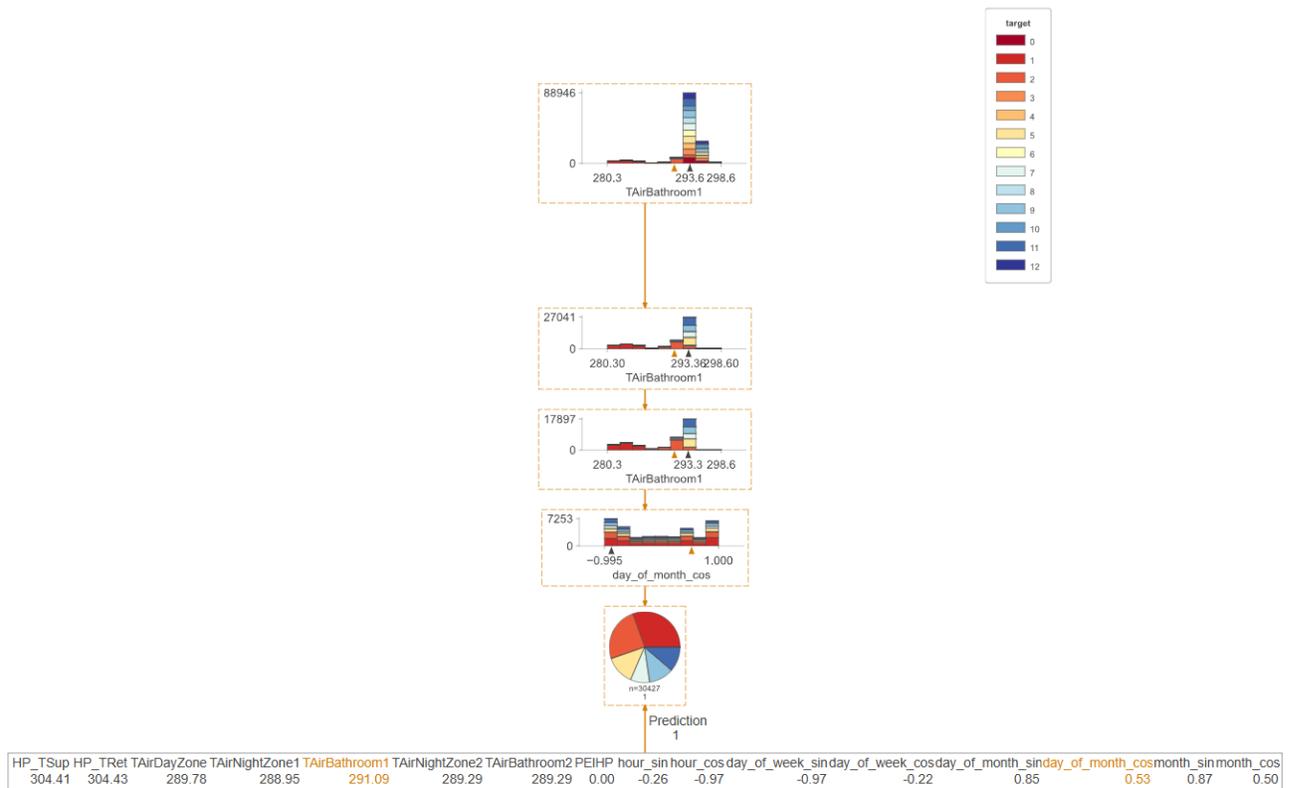


Figure 13: An example of a local prediction path. Full SVG figure can be found [here](#).

SHAP values are computed as the average of the marginal contributions across all possible coalitions of features. Mathematically, the Shapley value for a feature J is defined in Eq. (7).

$$\phi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(n - |S| - 1)!}{n!} [f(S \cup \{j\}) - f(S)] \quad (7)$$

Where N is the set of all features, S is a subset of features excluding j , $f(S)$ represents the prediction model's output using the features in set S , and n is the total number of features. This formula calculates the average impact of including the feature j in the model across all possible combinations of other features.

In the context of an XGBoost model used for either classification or regression tasks, SHAP can provide both global and local interpretability. Globally, SHAP values can reveal the overall importance of features across all predictions, showing which features are generally most influential for the model's decisions. Locally, SHAP offers detailed insights into why individual predictions were made, attributing specific output changes to particular features for single instances.

Regarding the global interpretation, the average impact of each feature on certain predicted class can be estimated by aggregating the SHAP values as shown in Figure 14. Confirming the previous results, Bathroom 1, Living room and the Bedroom 1 temperatures have the highest impact on the predictions.

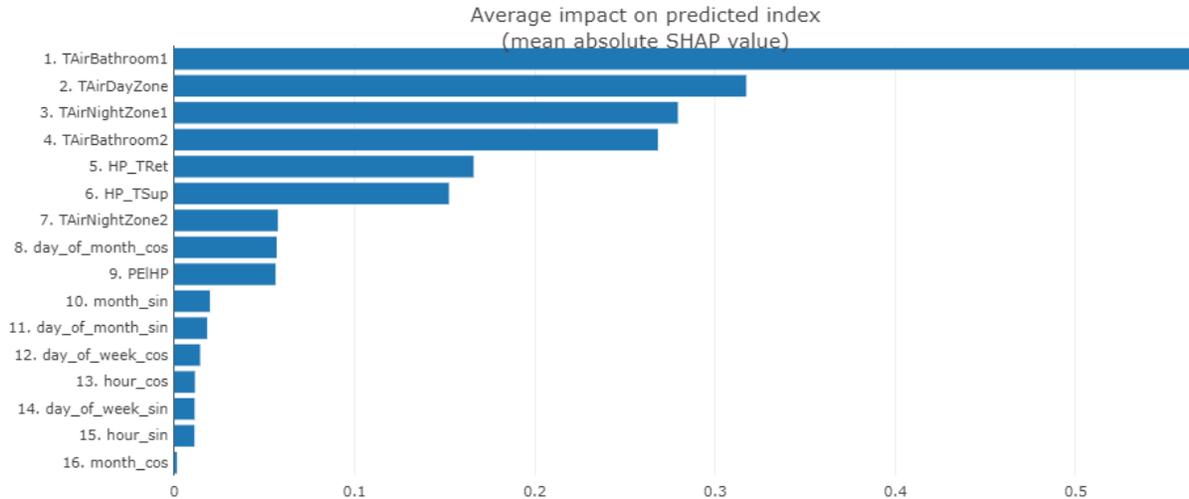


Figure 14 Global interpretation of the model using aggregated SHAP value.

Also, the feature importance for the prediction for each individual class can be plotted using the mean absolute SHAP value for the specific class as shown in Figure 15. The Figure displays the feature importance for Class 10 (constant noise added to the living room temperature sensor). The x axis represents the SHAP values for this class. Higher positive values increase the likelihood of predicting Class 8 and negative values decrease the likelihood. The features are ordered by important. The bathroom temperature sensor has the highest importance followed by the living room temperature. As for the spread in values for each feature, the wider the spread of points, the more varied the feature's impact across different instances. The coloured scale indicates the values of the features. As expected for example, the higher the temperature of the living room the more positive effect it has on increasing the likelihood of Class 8.

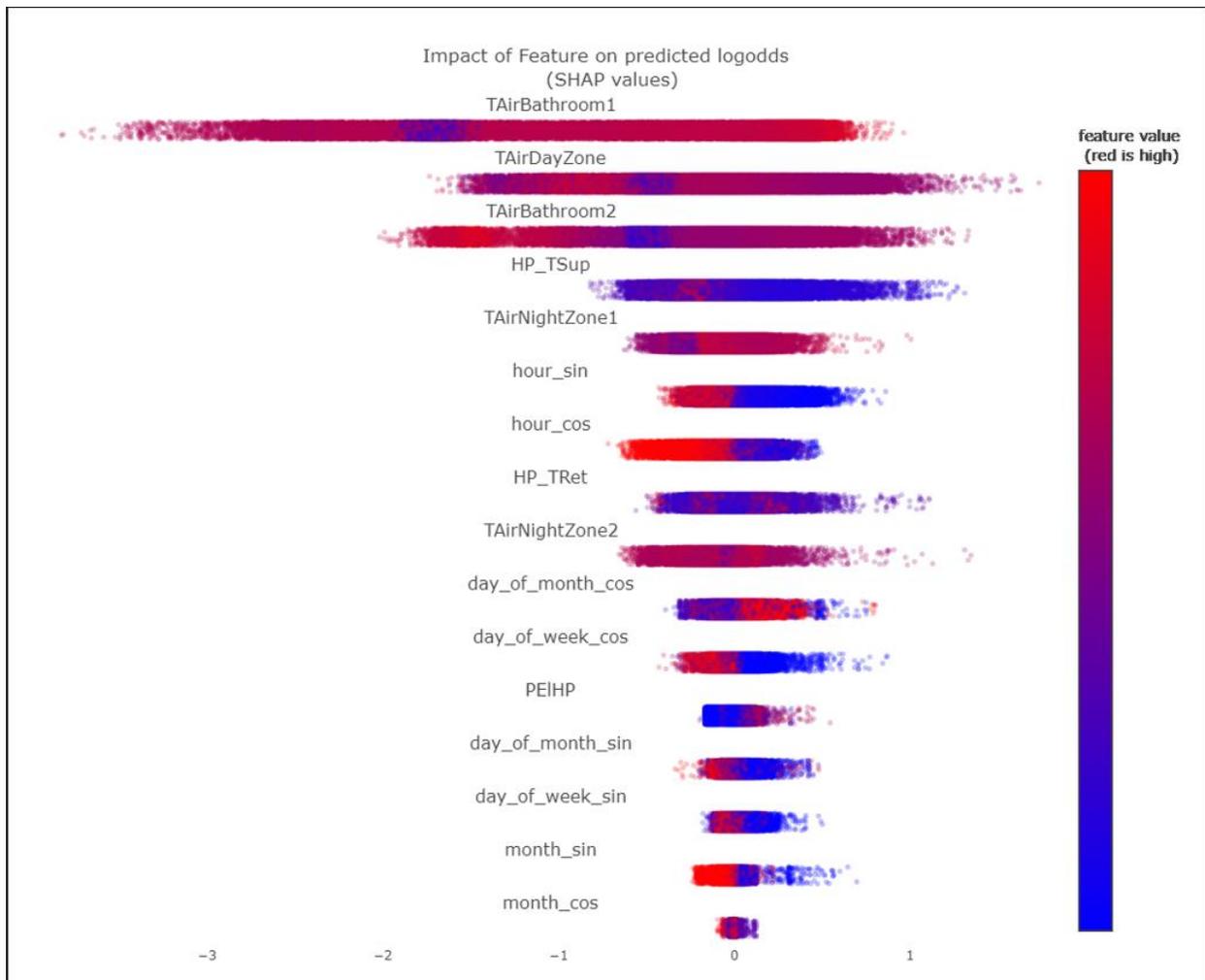


Figure 15: Features importance for predicting Class 8 (constant noise on living room sensor).

Locally for each prediction, the contribution of each feature to the prediction can be extracted as shown in Figure 16. The figure shows one point that was predicted as Class 8 and the contribution of each feature to the SHAP value. The green bars represent positive effect on predicting Class 8 (the right prediction in this case), while the red bars mean the features contributes negatively for the prediction of the class.

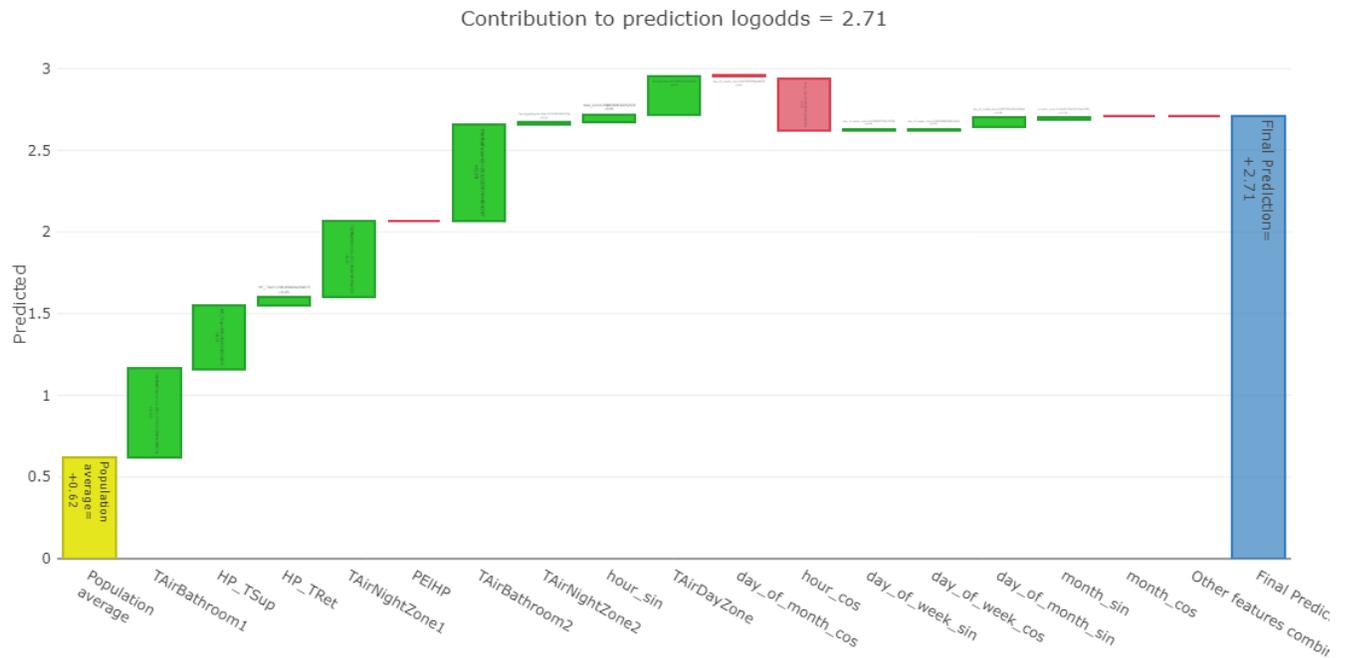


Figure 16 Local explanation of one prediction of Class 8 using SHAP.

It is worth noting that also the probability of each class for each prediction can be quantified. It was notable that the classes that are poorly predicted like Class 0, 3, 4 and 6 have almost equal probabilities between multiple classes meaning that usually the model is not confident about the decisions of those classes. To demonstrate this, Figure 17 and Figure 18 are two examples of individual predictions. In the first example in Figure 17, the model has almost split decision between class 3 and 4 reflecting high uncertainty in the prediction. While in the second example in Figure 18, the model misclassified the point as class three while the actual class was zero. It is also visible that the model was highly uncertain about the prediction.

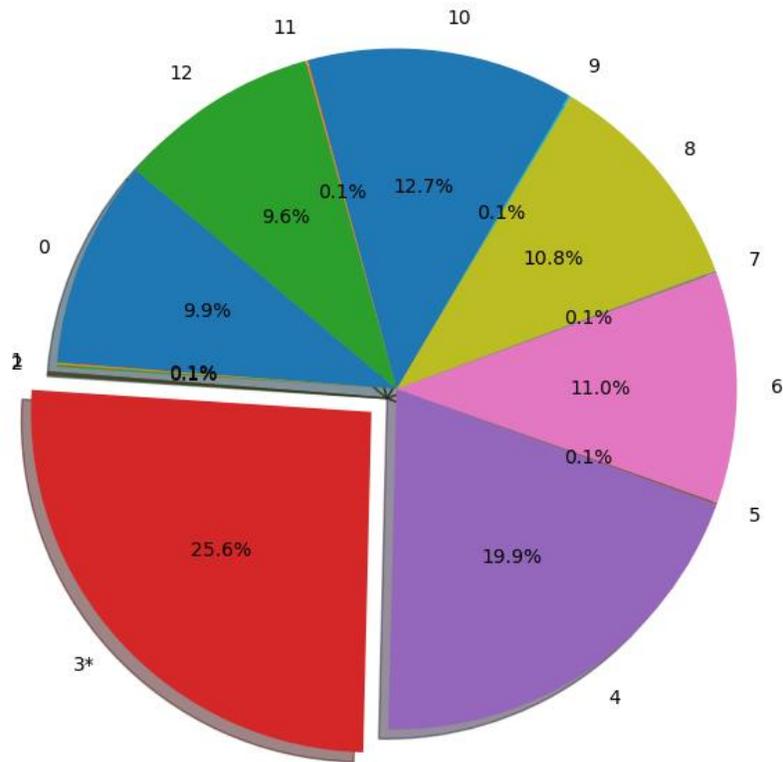


Figure 17: Local prediction probabilities. The actual class is Class 3 (highlighted), and the model predicted it.

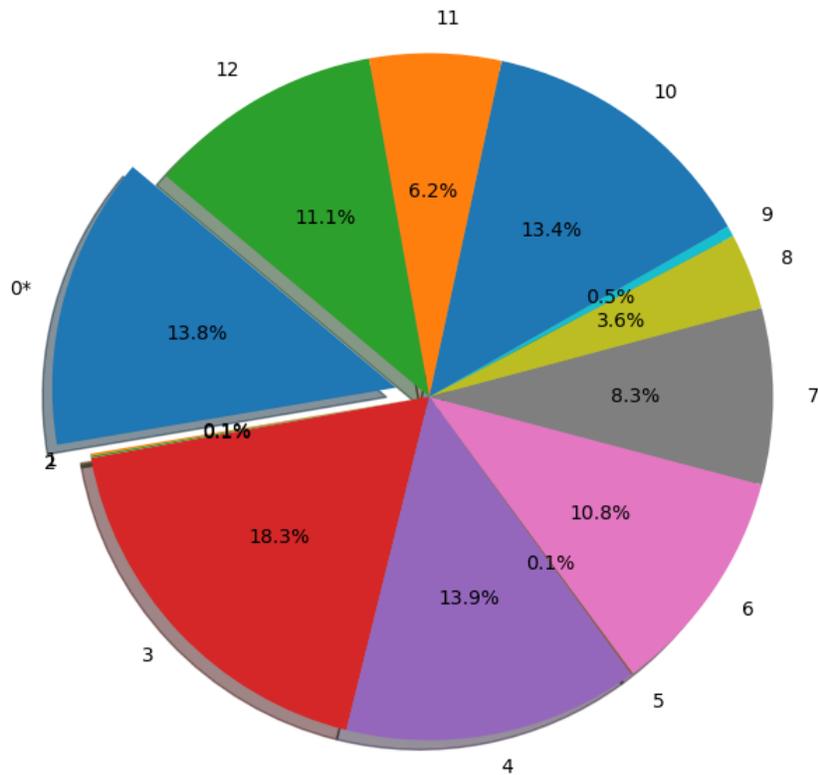


Figure 18: Local prediction probabilities. The actual class is Class 0 (highlighted), and the model predicted Class 3.

3 Multivariate time series classification algorithms benchmarking

The goal of this chapter is to investigate the use of 9 algorithms that are designed specifically for multivariate time series classification. The algorithms will be tested against three open-source datasets used for benchmarking FDD algorithms. The problem is formulated as multi class classification task. This chapter is divided into three sections. Firstly, the algorithms used will be explained, then the datasets used, and the data preprocessing procedure will be illustrated. Lastly the results will be discussed.

3.1 Background

This section is organized into six subsections. The first subsection provides an overview of time series definitions and their relevance in FDD research. The subsequent five subsections each focus on a different type of time series classifier: distance-based, interval-based, convolutional-based, deep learning-based, and dictionary-based.

3.1.1 Time series definition and status in FDD research

In the classification of univariate time series, each example is defined by a pair (x, y) , where x consists of m observations (x_1, \dots, x_m) representing the time series, and y is a categorical outcome with a number of possible outcomes. Classification algorithms predict the probability distribution of y based on x . In the context of Multivariate Time Series Classification (MTSC), the data input is a collection of vectors within a d dimensional space denoted as $x = (x^1, \dots, x^d)$ with each x_k representing a sequence $(x_{1,k}, x_{2,k}, \dots, x_{m,k})$. We reference the j -th data point in the i -th sequence as $x_{i,j,k}$. MTSC presents unique challenges due to the possibility of relevant features stemming

from the interplay between different dimensions, in addition to the autocorrelation within a single time series. Moreover, the substantial amount of data can hinder the identification of distinctive features. MTSC algorithms can be sorted into similar categories as univariate Time Series Classification (TSC) methods, based on their foundational approach, including methods based on distance metrics, shapelets, histograms, and neural networks. This section provides a brief explanation of the eight different algorithms employed in this study, categorized into five distinct types: distance-based, interval-based, deep learning-based, convolutional-based, and dictionary-based methods.

As demonstrated in the introductory chapter, many studies have shown the effectiveness of the data driven approach for FDD applications in buildings HVAC systems. However, few studies have tried to deal with the temporal dependence in HVAC systems operational data or benchmark algorithms that natively with time series. Wang et al. [98] introduced an innovative hybrid FDD approach for chillers. This approach leveraged time series data and integrated a 1D-CNN with a GRU. Their findings demonstrated that this method was particularly effective, especially in identifying minor faults, outperforming other techniques like 1D-CNN, GRU, LSTM, BPNN, PCA_BPNN, and 1D-CNN_LSTM. Jiang et al. [99] created a modular framework incorporating machine learning techniques for supervised learning on time series data. This framework combined an autoencoder (AE) with classifiers to predict faults in chillers using time series data. The results showed that the framework substantially improved prediction accuracy, achieving around a 30% increase over baseline models.

3.1.2 Distance-based classifiers

Distance-based classifiers measure the similarity between time series using specialized distance functions tailored for time series data. Often referred to as elastic distances, these functions are designed to adjust for misalignments between series by allowing for shifts or modifications within the series. A highly favoured method for Time Series Classification (TSC) involves the application of a 1-Nearest Neighbour classifier [14], integrated with a specialized distance function. This function is designed to correct for any misalignment in the series by enabling adjustments. The most widely used distance function for this task is Dynamic Time Warping (DTW). To compute the distance between two time series, one designated as $a = (a_1, a_2, \dots, a_m)$ and the other as $b = (b_1, b_2, \dots, b_n)$, the following procedure is adopted:

1. Construct a matrix M of dimensions $m \times n$ where the element $M_{i,j}$ represents the square of the difference between a_i and b_j ;
2. A warping path P , which is a sequence of matrix elements $P = ((e_1, f_1), (e_2, f_2), \dots, (e_s, f_s))$, is selected from matrix M in such a way that it follows specific rules:
 - The path starts at $(e_1, f_1) = (1,1)$ and ends at $(e_s, f_s) = (m, n)$;
 - The path increments are constrained such that $0 \leq e_{i+1} - e_i \leq 1$ for all i and $0 \leq f_{i+1} - f_i \leq 1$ for all i .
3. The distance along the path P denoted as D_p is the sum of the elements of M along P , defined as $D_p = \sum_{i=1}^s M_{e_i, f_i}$;
4. The goal is to find P^* over all possible paths P through matrix M , thus $P^* = \min_{P \in D} D_p(a, b)$;
5. To determine the optimal distance, the following recursive relation is employed, which computes the minimum of three neighbouring points in the matrix:

$$DTW(i, j) = M_{i,j} + \min \begin{cases} DTW(i-1, j). \\ DTW(i, j-1). \\ DTW(i-1, j-1). \end{cases} \quad (8)$$

The final computed distance using DTW is $DTW(m, n)$ after applying the recurrence relation throughout the matrix. In this study, we used the 1 NN method with DTW distance function. For short the method will be referred to as DTW.

3.1.3 Interval based classifiers

Approaches based on intervals examine specific segments of the complete series that are phase-dependent, deriving aggregate statistics from these subsections to facilitate classification. We used the Canonical Interval Forest (CIF) [101] in this study. CIF combines the capabilities of Time series Forest (TSF) [102] and catch22 [17]. While TSF traditionally uses simple summary statistics (mean, standard deviation, slope) for each interval, CIF incorporates a set of 22 more complex and descriptive features from the "catch22" toolkit. These features cover various aspects of the time series data, providing a richer set of descriptors. CIF employs a forest of decision trees to classify time series based on the extracted features. Each tree in the forest makes a decision based on a

subset of features and intervals, and the final classification is determined by a majority vote across all trees in the forest. This ensemble approach helps improve accuracy and robustness.

3.1.4 Convolutional based classifiers

Convolution involves using a subsequence to extract features from a time series. This process involves sliding the convolution across the series and computing the dot product at each position. This generates a new series, commonly referred to as an activation map or feature map, where higher values indicate a strong correlation with the convolution.

ROCKET [104] derives two key features from the output feature maps: the highest value, often termed as a max pooling operation, and the ratio of positive values, also known as positive predictive value (PPV). For instance, considering the first element of the feature map, which is computed through a dot-product operation between $T_{1:3} * u = T_{1:3} \cdot u$ resulting in $0 + 0 + 3 = 3$. The max pooling method identifies the peak value from the feature map to be used as a feature, and in the given example, the PPV is calculated to be $8/11$. Numerous random convolutions are created and integrated with these two features to form an enhanced training dataset. This enriched dataset is then applied to train a linear classifier. ROCKET also incorporates dilation, which effectively acts as a down sampling mechanism by creating intervals between data points. In this context, a convolution with a dilation factor of d is matched against data points that are spaced d steps apart to measure the separation. ARSENAL [105] an ensemble of ROCKET transformers using Ridge classifier [106] base classifier. Weights each classifier using the accuracy from the ridge cross-validation. Allows for generation of probability estimates at the expense of scalability compared to ROCKET.

3.1.5 Deep learning-based classifiers

Wide range of neural network-based architectures have been used for TSC purposes in the literature [107]. In this study we used three deep learning-based classifiers.

In their 2017 paper, Wang et al. introduced an advanced Residual Network (ResNet) [108]. This model is 11 layers, with the initial nine being convolutional layers followed by a Global Average Pooling (GAP) layer that processes the time series data across the temporal dimension. ResNets are distinguished by their use of shortcut residual connections that link the output of a residual

block back to its input. This setup facilitates the direct flow of gradients during training, significantly mitigating the issue of vanishing gradients. The described network structure includes three such residual blocks, each capped by a GAP layer and culminating in a softmax classifier. The classifier is designed to have as many neurons as there are classes in the dataset. Within each block, the layers consist of three sequential convolutions, where the outputs are combined with the block's input and forwarded to the subsequent layer. Uniformly across the network, each convolution utilizes 64 filters, employs the ReLU activation function, and is preceded by a batch normalization step. The lengths of the filters in these convolutions vary, being 8, 5, and 3 for the first, second, and third convolutions, respectively.

The Multivariate Long Short-Term Memory Fully Convolutional Network (MLCN) [109] merges LSTM and FCN technologies to enhance multivariate time series classification. This architecture leverages LSTMs to learn sequence dependencies and FCNs for feature extraction. Additional adaptations include squeeze-and-excitation blocks in the first two convolutional layers to adjust feature map interdependencies. The model comes in two variants, one with and one without an attention mechanism in the LSTM layers. However, studies across 35 datasets revealed minimal performance differences between the two. For simplicity and reproducibility, the version without the attention mechanism is preferred. In original tests, the number of LSTM cells was variable; in current applications, it is fixed at 64 for consistency across datasets [110].

InceptionTime is an ensemble of deep convolutional neural network models tailored for TSC [111]. This approach leverages the architecture of Inception-v4 by deploying multiple Inception modules within each network, where each module applies a variety of filters simultaneously to the input time series. This allows the network to capture and learn from a broad range of features at different scales. To enhance stability and performance, InceptionTime combines five such networks with randomly initialized weights, utilizing the ensemble's aggregate output for classification. This structure provides a robust and efficient way to handle the complexity and diversity of time series.

3.1.6 Dictionary based classifiers

Dictionary based approaches adapt the bag of words model commonly used in signal processing, computer vision and audio processing for time series classification [112]. These approaches use phase-independent subsequence's by sliding a window over time series. However, rather than

measuring the distance to a subsequence, as in shapelets, each window is transformed into a word, and the frequency of occurrence of repeating patterns is recorded. Dictionary based methods usually involve several steps:

1. subseries Extraction: Each time series is divided into overlapping windows or subseries.
2. discretization: Each window is transformed into a discrete-valued word. This involves normalizing the values in the window to have a uniform standard deviation, reducing the dimensionality using a truncated Fourier transform to retain only the most significant coefficients, and then converting these coefficients into symbols from a fixed size alphabet.
3. feature Vector Construction: A sparse feature vector is created from histograms of the word counts.
4. classification: These feature vectors are then used with machine learning algorithms to classify the time series.
5. In this study, we use the MUSE method [113] which extends the WEASEL algorithm [114] to handle multivariate time series data, employing the Symbolic Fourier Approximation (SFA) for the discretization process. This method stands out by its ability to effectively transform real-valued measurements into discrete symbols, enabling sophisticated pattern recognition in complex datasets.

In this study, we use the MUSE method [113] which extends the WEASEL algorithm [114] to handle multivariate time series data, employing the Symbolic Fourier Approximation (SFA) for the discretization process. This method stands out by its ability to effectively transform real-valued measurements into discrete symbols, enabling sophisticated pattern recognition in complex datasets.

3.2 Datasets description and preparation

This section is divided into two subsections. First subsection is introducing the datasets used in the study including the faults implemented. The second subsection is describing the data preprocessing procedure.

3.2.1 Datasets description

Three datasets for fault detection and diagnostics from the Lawrence Berkley National Laboratory (LBNL) [115] have been used. These datasets can be used to evaluate and benchmark the performance accuracy of FDD algorithms or tools. It contains operational data from simulation and laboratory experiments. In this study we used three of the eight datasets. The three datasets used are generated using Modelica and EnergyPlus [116]. The simulations generated one year of data, each with a time step of one minute. Each dataset contains one file of fault free case and several files with different faults injected into the models. A brief description of the datasets and the faults imposed is given in the following subsection. The full description, data points definition and control sequences can be found in the documentation provided for each dataset in [115].

The first dataset is a simulated boiler plant -shown in Figure 19 - serving a 12-story building with individual floors having a dedicated AHU serving five zones, and individual zones having a dedicated VAV terminal unit. Each terminal unit has a reheat coil that uses hot water produced by the plant. The plant consists of two parallel boilers and pumps that distribute the hot water to these reheat coils. Sensors and valves are also used to control water flow through the plant. The dataset contains 22 features both continuous and discrete signals. The faults imposed in the model are provided in Table 5.

Table 5 Input scenarios and fault imposed in the boiler plant model. Taken from [106]

Input scenarios		Method of Fault Imposition
Fault type	Fault intensity	
The hot water leaving temperature sensor of boiler 1	Sensor bias	Add bias to sensor output
The hot water leaving temperature sensor of the hot water loop		
The differential pressure sensor in the hot water loop		
Boiler 1 heat exchanger	Fouling	Multiply intensity value by heat transfer coefficient
Controller PI for boiler supply temperature setpoint	Inappropriate tuning	Modify gain value of controllers

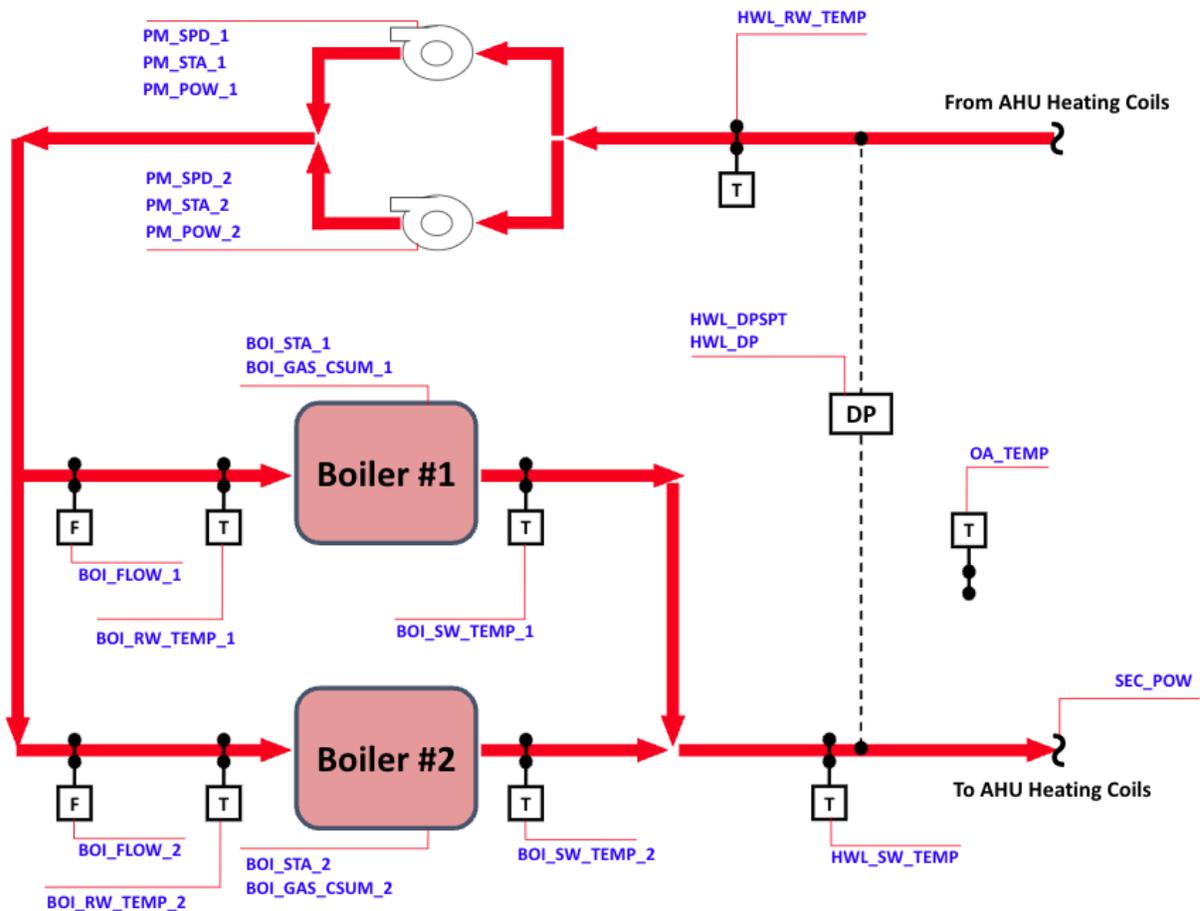


Figure 19: Schematic of the boiler plant taken from [106].

The datapoints used in the training process is shown in Table 6

Table 6 Data points of the boiler datasets used in training

Data point name	Diagram point Abbreviation	Description
Outdoor Air: Dry Bulb Temperature	OA_TEMP	Dry bulb temperature of outdoor air
Secondary Loop Load	SEC_POW	Calculated heating load from hot water loop, product of hot water loop flow and supply/return temperature difference
Hot Water Loop: Differential Pressure	HWL_DP	Pressure differential of the hot water loop
Hot Water Loop: Supply Water Temperature	HWL_SW_TEMP	Temperature of the water leaving the hot water loop
Hot Water Loop: Return Water Temperature	HWL_RW_TEMP	Temperature of the water entering the hot water loop
Hot Water Loop Supply Water Temperature Setpoint	HWL_DPSPT	Setpoint for temperature of the water leaving the hot water loop
For boiler 1 and 2 (the name of data points is followed by 1, 2, respectively):		
Boiler: Status	BOI_STA	On-off status of a boiler (0,1)
Boiler: Supply Water Temperature	BOI_SW_TEMP	Temperature of the water leaving a boiler
Boiler: Return Water Temperature	BOI_RW_TEMP	Temperature of the water entering a boiler
For each hot water pump (pump variables are followed by 1, 2, respectively):		
Hot Water Loop Pump: Speed Ratio	PM_SPD	Speed of a pump
Pump: Status	PM_STA	On-off status of a pump
Pump: Power Consumption	PM_POW	Power consumption of pump

The second dataset is simulated chiller plant serves the same building as the boiler plant shown in Figure 20. The chiller plant serves the dedicated AHU on each floor with cold water. The plant consists of a primary loop with three chillers for producing chilled water, a secondary loop for delivering chilled water to the AHUs, and a condenser water loop with cooling towers for rejecting heat to the ambient. Sensors, pumps, and valves are used to control water flow through the plant. The dataset contains 77 features both continuous and discrete signals. The faults imposed in the model are provided in Table 7.

Table 7: Input scenarios and fault imposed in the chiller plant model. Taken from [106].

Input scenarios		Method of Fault Imposition
Fault type	Fault intensity	

The chilled water leaving temperature sensor of Chiller 1	Sensor bias	-2°C, -1°C, 1°C, 2°C	Add bias to sensor output
The condenser water leaving temperature sensor of Cooling tower 1			
The differential pressure sensor in the secondary chilled water loop		-20%, -10%, 10%, 20%	
The condenser water leaving the three-way valve	Leakage	25%, 50%, 75%	Increase the default minimum position setting
The condenser water leaving the three-way valve	Stuck	50%, 75%	Assign a fixed simulated controlled device position
Cooling tower 1 heat exchanger	Fouling	95%, 80%, 65%	Multiply intensity value by heat transfer coefficient
Controller PI for condenser loop supply temperature setpoint	Inappropriate tuning	-	Modify gain value of controllers

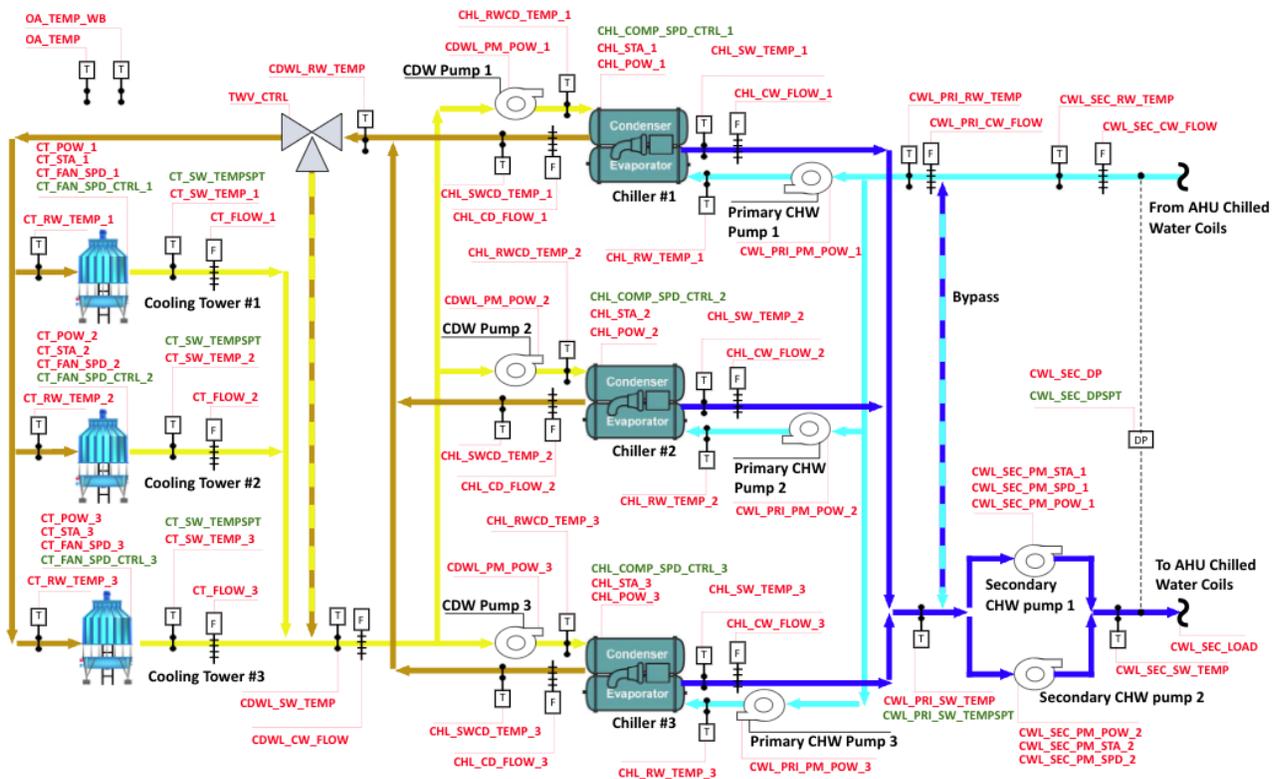


Figure 20 Schematic of the chiller plant taken from [106].

The data points used in the training process are shown in Table 8.

Table 8 Data points of the Chiller plant datasets used in training

Data point name	Diagram Point Abbreviation	Description
Secondary Loop Cooling Load Calc	CWL_SEC_LOAD	Calculated cooling load from secondary loop, product of secondary loop flow and supply/return temperature difference
Secondary Chilled Water Loop: Return Water Temperature	CWL_SEC_RW_TEMP	Temperature of the water entering the secondary chilled water loop
Secondary Chilled Water Loop: Supply Water Temperature	CWL_SEC_SW_TEMP	Temperature of the water leaving the secondary chilled water loop
Secondary Chilled Water Loop: Chilled Water Flow Rate	CWL_SEC_CW_FLOW	Flow rate of the secondary chilled water loop
Three-way Valve: Control Signal	TWV_CTRL	Control signal of condenser water loop 3-way mixing valve
Cooling Tower Supply Water Temperature Setpoint	CT_SW_TEMPSP T	Set point of temperature of the condenser water leaving cooling tower
Condenser Water Loop: Supply Water Temperature	CDWL_SW_TEMP	Temperature of the water leaving the condenser water loop
Condenser Water Loop: Return Water Temperature	CDWL_RW_TEMP	Temperature of the water entering the condenser water loop
Condenser Water Loop: Chilled Water Flow Rate	CDWL_CW_FLOW	Flow rate of the condenser water loop
Primary Chilled Water Loop: Return Chilled Water Temperature	CWL_PRI_RW_TEMP	Temperature of the water entering the primary chilled water loop

Primary Chilled Water Loop: Supply Water Temperature	CWL_PRI_SW_T EMP	Temperature of the water leaving the primary chilled water loop
Primary Chilled Water Loop: Chilled Water Flow Rate	CWL_PRI_CW_F LOW	Flow rate of the primary chilled water loop
Primary Loop Chilled Water Supply Temperature Setpoint	CWL_PRI_SW_T EMPSPT	Setpoint for temperature of the water leaving the primary chilled water loop
Outdoor Air: Dry Bulb Temperature	OA_TEMP	Dry bulb temperature of outdoor air
Outdoor Air: Wet Bulb Temperature	OA_TEMP_WB	Wet bulb temperature of outdoor air
Secondary Loop Differential Pressure Setpoint	CWL_SEC_DPSP TS	Setpoint of Secondary loop differential pressure
Secondary Chilled Water Loop: Pressure Differential	CWL_SEC_DP	Pressure differential of the secondary chilled water loop
For cooling tower 1, 2, 3 , the name of data points is followed by 1, 2, 3, respectively		
Cooling Tower: Status	CT_STA	On-off status of a cooling tower
Cooling Tower: Water Flow Rate	CT_FLOW	Flow rate of a cooling tower
Cooling Tower: Return Water Temperature	CT_RW_TEMP	Temperature of the water entering a cooling tower
Cooling Tower: Supply Water Temperature	CT_SW_TEMP	Temperature of the water leaving a cooling tower
Cooling Tower: Speed	CT_FAN_SPD	Speed of a cooling tower fan
Cooling Tower: Speed Control Signal	CT_FAN_SPD_C TRL	Control signal for cooling tower fan speed
Cooling Tower: Power Consumption	CT_POW	Power consumption of a cooling tower

For chiller 1, 2, 3 , the name of data points is followed by 1, 2, 3, respectively:		
Chiller: Status	CHL_STA	On-off status of a chiller
Chiller: Control Signal	CHL_COMP_SPD _CTRL	Control signal for chiller compressor speed
Chiller: Chilled Water Flow Rate	CHL_CW_FLOW	Flow rate of the chilled water leaving a chiller
Chiller: Condenser Water Flow Rate	CHL_CD_FLOW	Flow rate of the condenser water leaving a chiller
Chiller: Return Chilled Water Temperature	CHL_RW_TEMP	Temperature of the chilled water entering a chiller
Chiller: Supply Chilled Water Temperature	CHL_SW_TEMP	Temperature of the chilled water leaving a chiller
Chiller: Supply Condenser Water Temperature	CHL_SWCD_TE MP	Temperature of the condenser water leaving a chiller
Chiller: Return Condenser Water Temperature	CHL_RWCD_TE MP	Temperature of the condenser water entering a chiller
Chiller: Power Consumption	CHL_POW	Power consumption of a chiller

The third dataset is generated from a simulated system consisting of a single-duct air handling unit (SD-AHU) providing conditioned air to five VAV terminal units, each serving a single zone (four perimeter and one interior) on the middle floor of a three-story building. The schematic of the system is shown in Figure 21. The SD-AHU has a chilled water-cooling coil, variable speed supply and return fans, and delivers air at a constant temperature and static pressure to the terminal units. Individual terminal units control the volume of air entering a zone and use hydronic reheat when necessary to satisfy the temperature setpoint in a zone. The dataset contains 30 features both continuous and discrete signals. The faults imposed in the model are provided in Table 9.

Table 9: Input scenarios and fault imposed in single duct AHU model. Taken from [106].

Input scenarios		Method of Fault Imposition
Fault type	Fault intensity	
Outdoor air temperature sensor bias	2°C, 4°C, -2°C, -4°C	Add bias to sensor output.
Supply air temperature sensor bias	2°C, 4°C, -2°C, -4°C	Add bias to sensor output.
Stuck outdoor air Damper	10%, 25%, 75%, 100% open	Automated override of outdoor air damper position to indicate that OA damper is stuck.
Leaking cooling coil valve	10%, 25%, 40%, 50%	Adjusted the minimum coil valve position value when the control signal is zero.
Stuck cooling coil valve	10%, 25%, 50%, 75%	Automated override of coil valve position to indicate that valve is stuck.

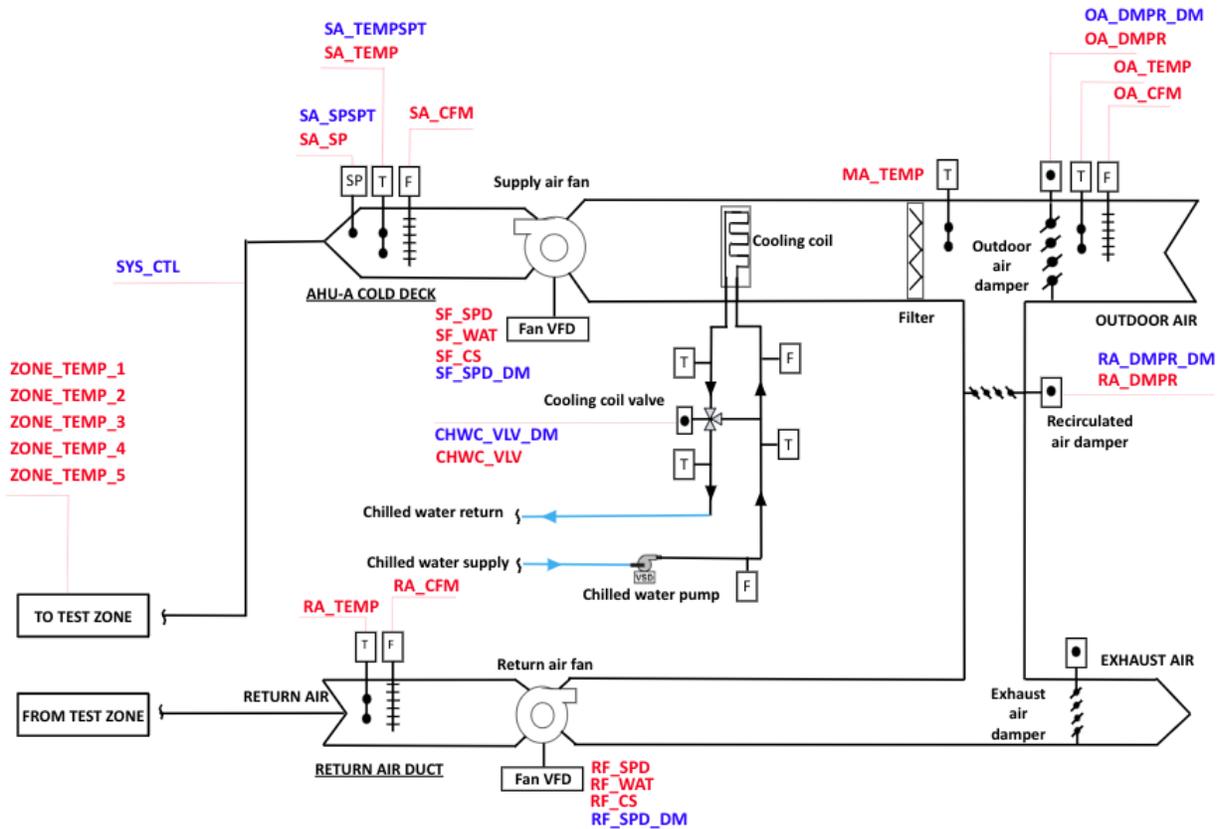


Figure 21: Schematic of the single duct AHU system taken from [106].

The data points used in the training process are shown in Table 10.

Table 10 Data points of the Single duct AHU datasets used in training

Data point name	Diagram Point Abbreviation	Description
AHU: Supply Air Temperature	SA_TEMP	AHU supply air temperature
AHU: Supply Air Temperature Set Point	SA_TEMPSP	AHU supply air temperature setpoint
AHU: Outdoor Air Temperature	OA_TEMP	AHU outdoor air temperature
AHU: Mixed Air Temperature	MA_TEMP	AHU mixed air temperature
AHU: Return Air Temperature	RA_TEMP	AHU return air temperature
AHU: Supply Air Fan Status	SF_SPD_DM	AHU supply air fan status; 0-off, 1-on
AHU: Return Air Fan Status	RF_SPD_DM	AHU return air fan status; 0-off, 1-on
AHU: Outdoor Volumetric Airflow	OA_CFM	AHU outdoor airflow
AHU: Return Volumetric Airflow	RA_CFM	AHU return airflow
AHU: Supply Volumetric Airflow	SA_CFM	AHU supply airflow
AHU: Supply Air Fan Speed Control Signal	SF_CS	Control signal for AHU supply air fan speed; ranges from 0 to 1; 0 - fan speed is 0%, 1 - fan speed is 100%
AHU: Supply Air Fan Speed Position	SF_SPD	AHU supply air fan speed; ranges from 0 to 1; 0 - fan speed is 0%, 1 - fan speed is 100%
AHU: Return Air Fan Speed Control Signal	RF_CS	Control signal for AHU return air fan speed; ranges from 0 to 1; 0 - fan speed is 0%, 1 - fan speed is 100%
AHU: Return Air Fan Speed Position	RF_SPD	AHU return air fan speed; ranges from 0 to 1; 0 - fan speed is 0%, 1 - fan speed is 100%
AHU: Supply Air Fan Power	SF_WAT	AHU supply air fan power
AHU: Return Air Fan Power	RF_WAT	AHU return air fan power
AHU: Outdoor Air Damper Control Signal	OA_DMPR_DM	Control signal for AHU outdoor air damper; ranges from 0 to 1; 0 – damper should be fully closed, 1 – damper should be fully open
AHU: Outdoor Air Damper Position	OA_DMPR	AHU outdoor air damper position; ranges from 0 to 1; 0 – damper should be fully closed, 1 – damper should be fully open
AHU: Return Air Damper Control Signal	RA_DMPR_DM	Control signal for AHU return air damper; ranges from 0 to 1; 0 – damper should

		be fully closed, 1 – damper should be fully open
AHU: Return Air Damper Position	RA_DMPR	AHU return air damper position; ranges from 0 to 1; 0 – damper should be fully closed, 1 – damper should be fully open
AHU: Cooling Coil Valve Control Signal	CHWC_VLV_DM	Control signal for AHU cooling coil valve; ranges from 0 to 1; 0 – valve should be fully closed, 1 – valve should be fully open
AHU: Cooling Coil Valve Position	CHWC_VLV	AHU cooling coil valve position; ranges from 0 to 1; 0 – valve should be fully closed, 1 – valve should be fully open
AHU: Supply Air Duct Static Pressure	SA_SP	AHU supply air duct static pressure
AHU: Supply Air Duct Static Pressure Set Point	SA_SPSPT	AHU supply air duct static pressure setpoint
Occupancy Mode Indicator	SYS_CTL	Indicator if the system operates in occupied mode; 1-occupied mode, 0-unoccupied mode
Zone 1: Air Temperature	ZONE_TEMP_1	Zone 1 Air Temperature
Zone 2: Air Temperature	ZONE_TEMP_2	Zone 2 Air Temperature
Zone 3: Air Temperature	ZONE_TEMP_3	Zone 3 Air Temperature
Zone 4: Air Temperature	ZONE_TEMP_4	Zone 4 Air Temperature
Zone 5: Air Temperature	ZONE_TEMP_5	Zone 5 Air Temperature

3.2.2 Datasets preparation

The procedure for the data preparation is the same for the three datasets. The data was first resampled from one-minute interval to ten minutes. This was done to reduce the computational cost after encountering multiple memory issues dealing with the data in the one-minute interval form. The resampling was done by means for the continuous variables and the mode for the discrete ones. Since the data was provided in a separate file for each fault state, the data was labelled, concatenated, and then normalized. All the TSC models need to have a specific format of the data of multi-index format, instances which is a consistent window of time and time points which is the timestep. We choose to prepare the data in a daily instance. The data was split into five folds each with an expanding number of instances for the training set while shifting the window of the testing set as demonstrated in Figure 22.

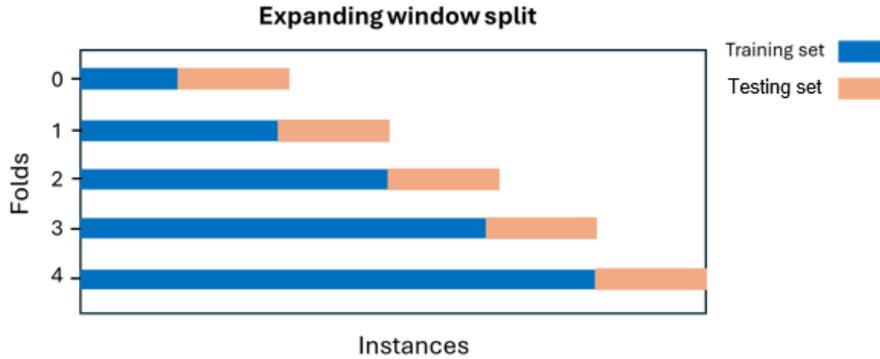


Figure 22: Expanding window for data splitting.

3.3 Results

This section includes four subsections. The first evaluates the performance of the algorithms. The second analyse the computational load of each algorithm and the third subsection compare the classifiers type according to the types introduced in section 1. Finally, a comparison with an ensemble supervised algorithm used in chapter 2 is detailed.

3.3.1 Performance evaluation

Table 11 provides the aggregated results from the five-fold cross-validation applied across three datasets. The results include accuracy, precision, recall and F1 score. Each metric is defined in function of True or False positives (TP, FP) and True or False negatives (TN, FN). In Figure 23 the relative performance of the algorithms according to the F1 score is displayed and in Figure 24 the individual performance of each algorithm is shown.

The analysis of classifier performance across three datasets (Boiler Operation, Chiller Plant, and Single Duct AHU) reveals significant variations in accuracy and overall effectiveness. As shown in Figure 23 and can be deducted from Table 11, the LSTMFCN emerges as the top performer, consistently demonstrating superior performance across all datasets. On the Boiler Operation dataset, it achieves the highest accuracy (99.16%) and F1 score (99.16%), setting a benchmark for performance. This classifier's strong showing across all datasets indicates its robustness and adaptability to different types of time series data in HVAC systems. The CanonicalInterval Forest and KNeighborsTimeSeries also exhibit impressive performance, particularly on the Boiler Operation dataset, where they match or closely trail the LSTMFCN in accuracy and F1 score.

A notable trend observed is the general decline in performance for most classifiers when applied to the Single Duct AHU dataset. This suggests that this particular dataset may present unique challenges or complexities that are not as prevalent in the Boiler Operation or Chiller Plant datasets. For instance, while the LSTMFCN maintains the highest accuracy (66.07%) on the Single Duct AHU dataset, this is a significant drop from its performance on the other two datasets. Similarly, other top-performing classifiers like CanonicalInterval Forest and ResNet see their accuracy decrease to around 65-67% on this dataset. This consistent drop in performance across classifiers highlights the importance of considering dataset-specific characteristics when selecting and implementing classification models for HVAC systems. In the appendix at the end of the thesis, I investigate this performance drop in detail.

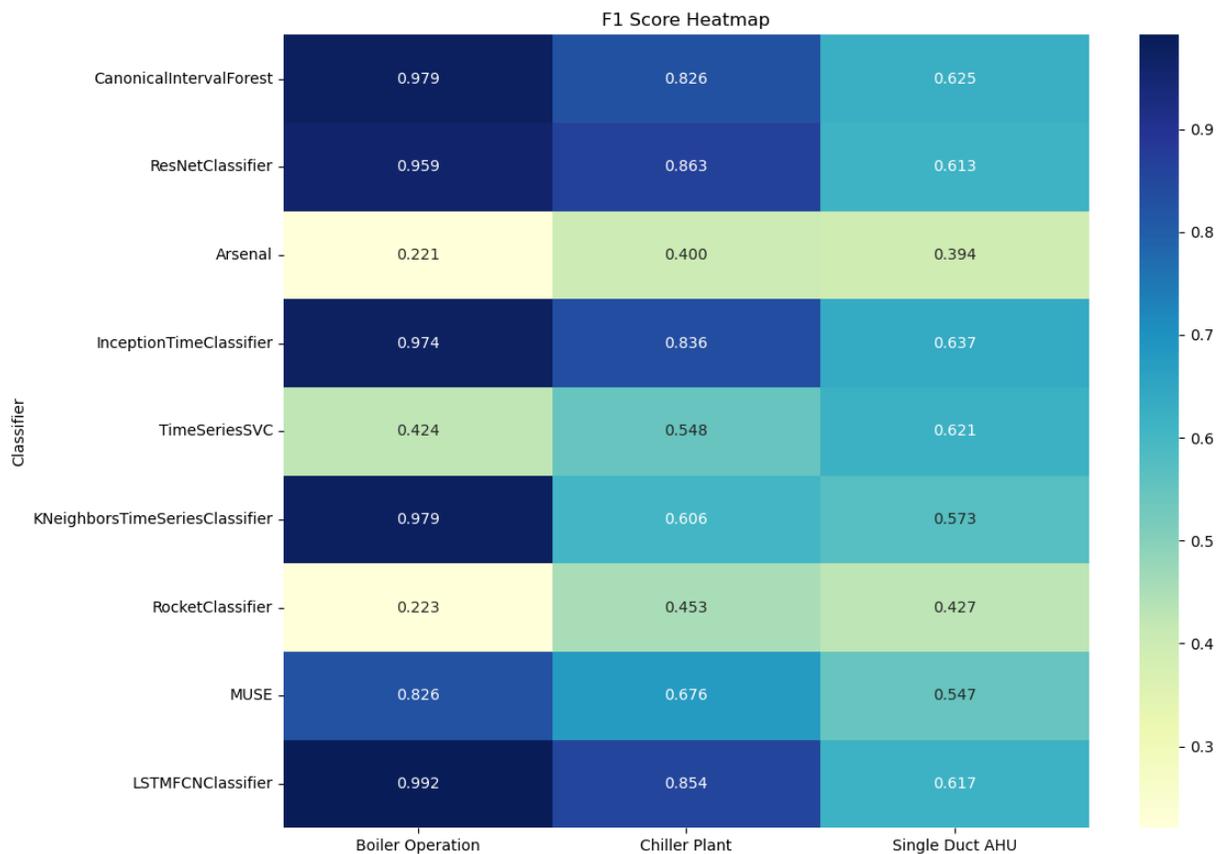
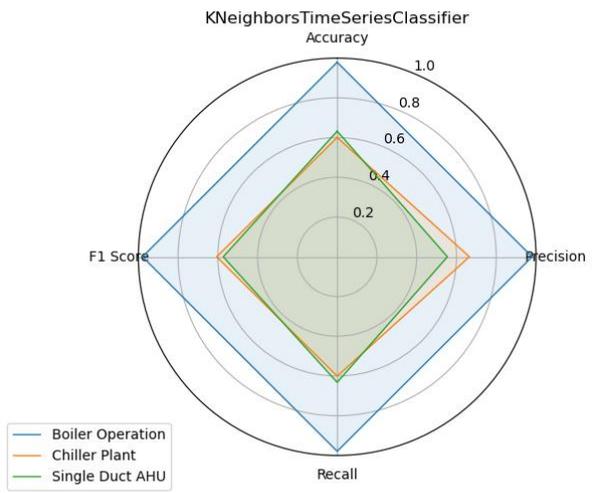
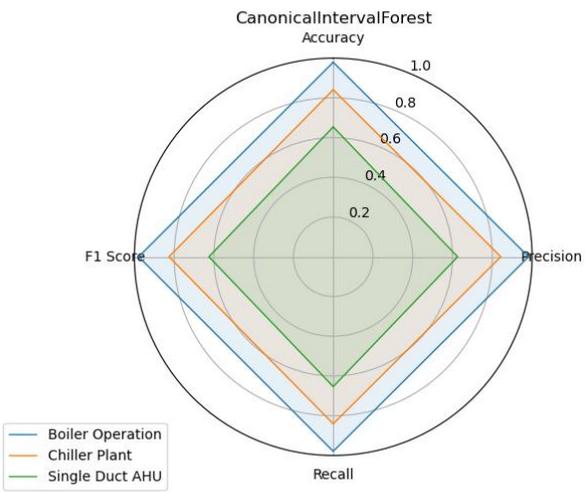
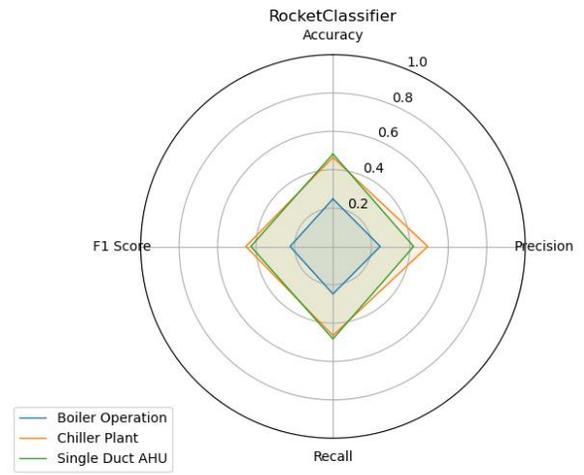
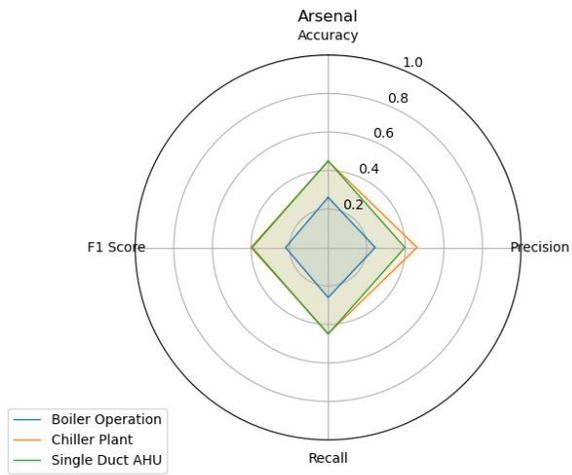
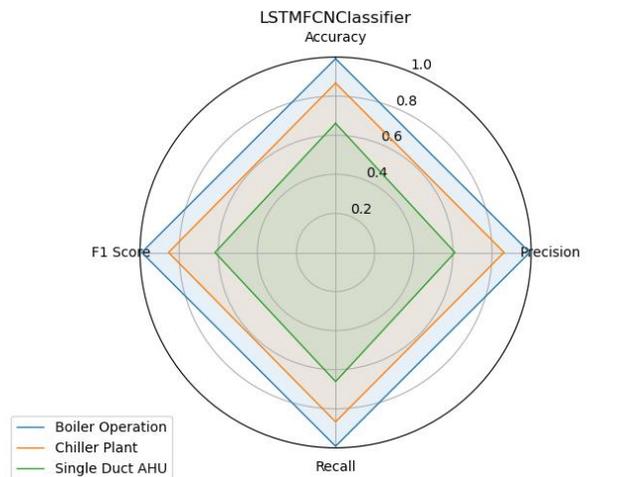
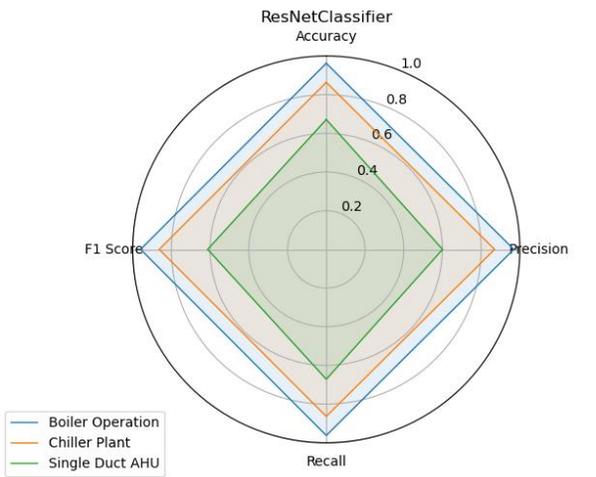
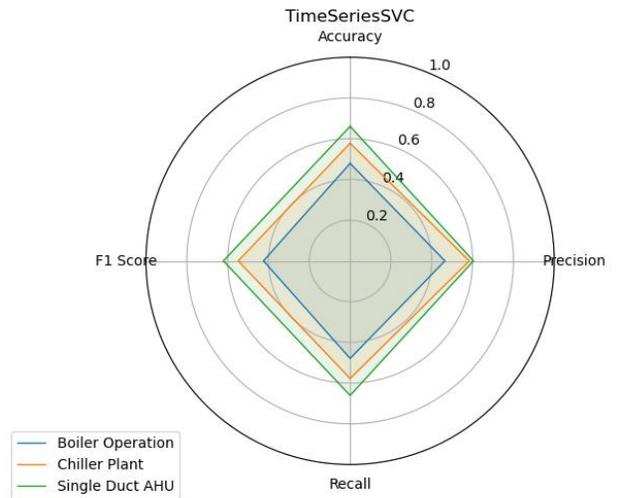
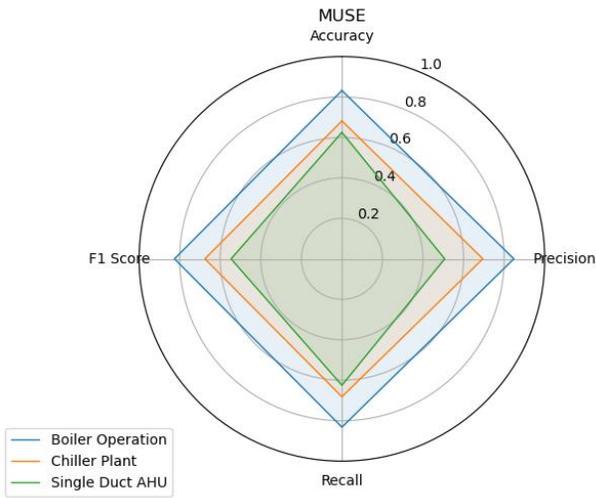


Figure 23: Heat map of F1 scores of all algorithms across datasets.

Some classifiers exhibit notably poor performance across all datasets. Arsenal and Rocket, for example, consistently show low accuracy and F1 score, with their best performance still falling

below 50% accuracy on any dataset. This can be clearly seen in their individual radar plots in Figure 24 and their relative performance in Figure 23. This suggests that these classifiers may not be well-suited for the specific challenges presented by HVAC system data, regardless of the particular subsystem being analysed. On the other hand, classifiers like InceptionTime and ResNet show moderate to good performance with an average accuracy of 0.81 and 0.82 across the three datasets respectively. Their performance, while not matching the top classifiers, remains consistent across datasets, indicating a certain level of reliability. These observations underscore the importance of thorough benchmarking and careful classifier selection in time series classification tasks for HVAC systems, as the choice of classifier can significantly impact the accuracy and reliability of the results.





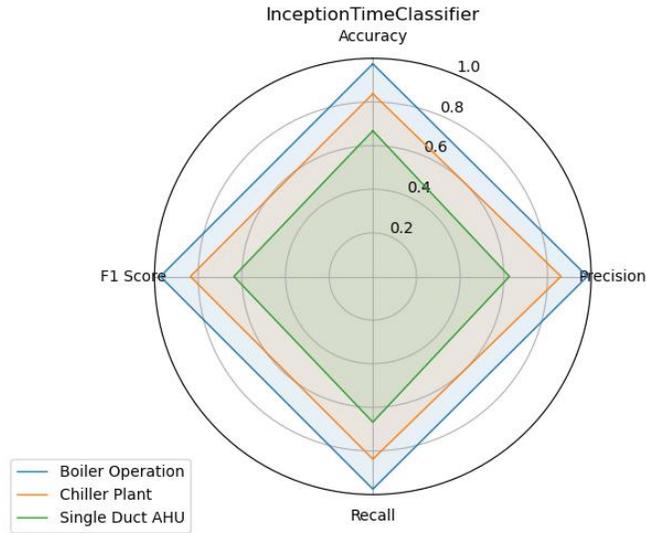


Figure 24: Radar plot for each algorithm performance on all three datasets.

Table 11 Results for the three datasets.

Classifier	SD-AHU					Chiller plant					Boiler plant				
	Runtime (min)	Accuracy	Precision	Recall	F1	Runtime (min)	Accuracy	Precision	Recall	F1	Run time (min)	Accuracy	Precision	Recall	F1
CIF	54	0.65	0.63	0.65	0.63	152	0.84	0.84	0.84	0.83	94	0.97	0.98	0.97	0.97
ResNet	500	0.67	0.60	0.67	0.61	558	0.86	0.87	0.86	0.86	385	0.96	0.97	0.96	0.96
Arsenal	10	0.45	0.40	0.45	0.39	10	0.45	0.46	0.45	0.40	12	0.26	0.24	0.26	0.22
InceptionTime	196	0.67	0.63	0.67	0.64	220	0.84	0.86	0.84	0.84	170	0.97	0.98	0.97	0.97
TimeSeriesSVC	123	0.66	0.60	0.66	0.62	175	0.58	0.58	0.58	0.55	85	0.48	0.46	0.48	0.42
1nn DTW	0.61	0.63	0.55	0.63	0.57	2.1	0.60	0.67	0.60	0.61	0.28	0.97	0.98	0.97	0.97
Rocket	60	0.48	0.42	0.48	0.43	68	0.46	0.49	0.46	0.45	54	0.24	0.25	0.25	0.22
MUSE	11.5	0.63	0.51	0.63	0.55	26	0.68	0.70	0.68	0.68	3	0.83	0.85	0.83	0.82
LSTMFC NC	120	0.66	0.61	0.66	0.62	232	0.87	0.86	0.87	0.85	98	0.98	0.98	0.98	0.98

3.3.2 Runtime analysis

Figure 25 illustrates the computational efficiency of each classifier and in Figure 26 the relative runtimes are shown for all classifiers across datasets are shown in form of a heatmap for an easier comparison. The runtime analysis of the classifiers across the three reveals significant variations in computational efficiency. The KNeighborsTimeSeries classifier consistently demonstrates the fastest performance, with runtimes ranging from a mere 0.29 minutes on the Boiler Operation dataset to 2.18 minutes on the Chiller Plant dataset. This exceptional speed makes it an attractive option for real-time or resource-constrained applications. In stark contrast, the ResNet classifier exhibits the longest runtimes across all datasets, requiring 384.87 minutes (about 6.4 hours) for the Boiler Operation dataset, 558 minutes (about 9.3 hours) for the Chiller Plant dataset, and 500 minutes (about 8.3 hours) for the Single Duct AHU dataset. This substantial computational demand suggests that ResNet may be more suitable for offline analysis or scenarios where computational resources are not a limiting factor.

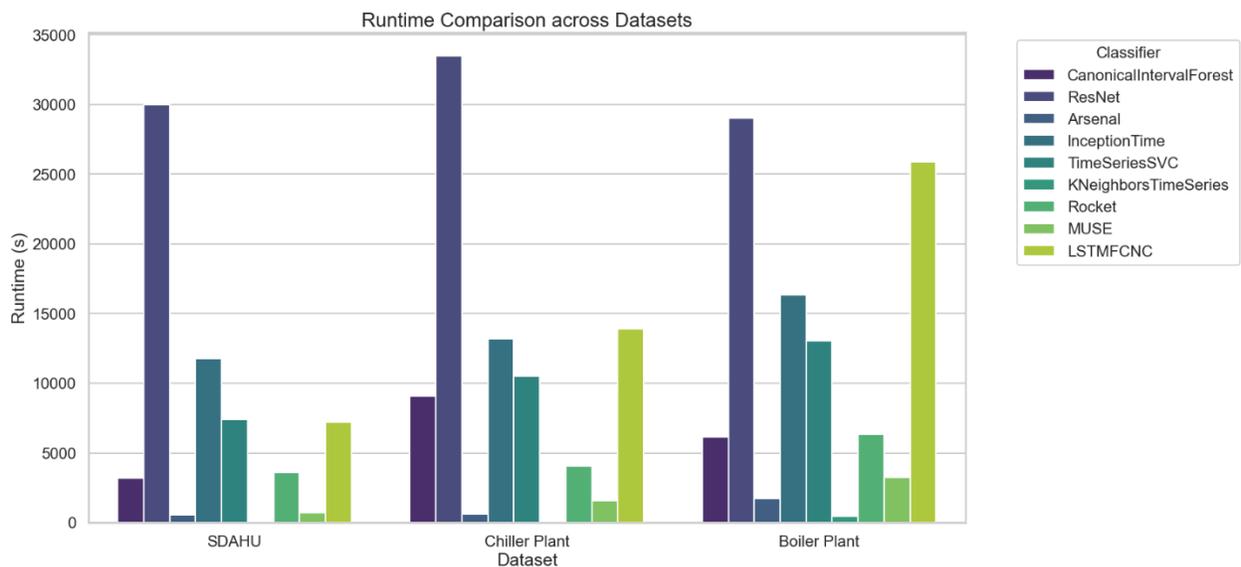


Figure 25: Runtime comparison across datasets.

Among the other classifiers, there is a wide spectrum of runtime performance. The LSTMFCN and InceptionTime classifiers, while showing high accuracy, also demand significant computational resources. LSTMFCN requires 97.9 minutes for the Boiler Operation dataset, 231.6 minutes for the Chiller Plant dataset, and 119.8 minutes for the Single Duct AHU dataset. InceptionTime shows similar patterns, with runtimes of 170.2, 220.1, and 196.01 minutes for the respective datasets. The

CanonicalIntervalForest classifier falls in the middle range, with runtimes of 94.2, 151.8, and 53.7 minutes across the three datasets. These moderate runtimes, combined with its good accuracy, may make it a balanced choice for many applications.

Interestingly, some classifiers that showed poor accuracy perform relatively quickly. The Arsenal classifier, for instance, completes its computations in 12.38, 9.91, and 9.52 minutes for the Boiler Operation, Chiller Plant, and Single Duct AHU datasets, respectively. Similarly, the Rocket classifier finishes in 53.4, 68, and 60.1 minutes across the three datasets. The MUSE classifier also shows relatively fast performance, with runtimes of 3.1, 25.9, and 11.6 minutes. However, the TimeSeriesSVC classifier, despite its convolutional nature, requires substantial computational time, with runtimes of 84.8, 174.9, and 123.4 minutes across the datasets. These results highlight the important trade-off between computational efficiency and classification accuracy, emphasizing the need to consider both factors when selecting a classifier for specific HVAC system analysis tasks.

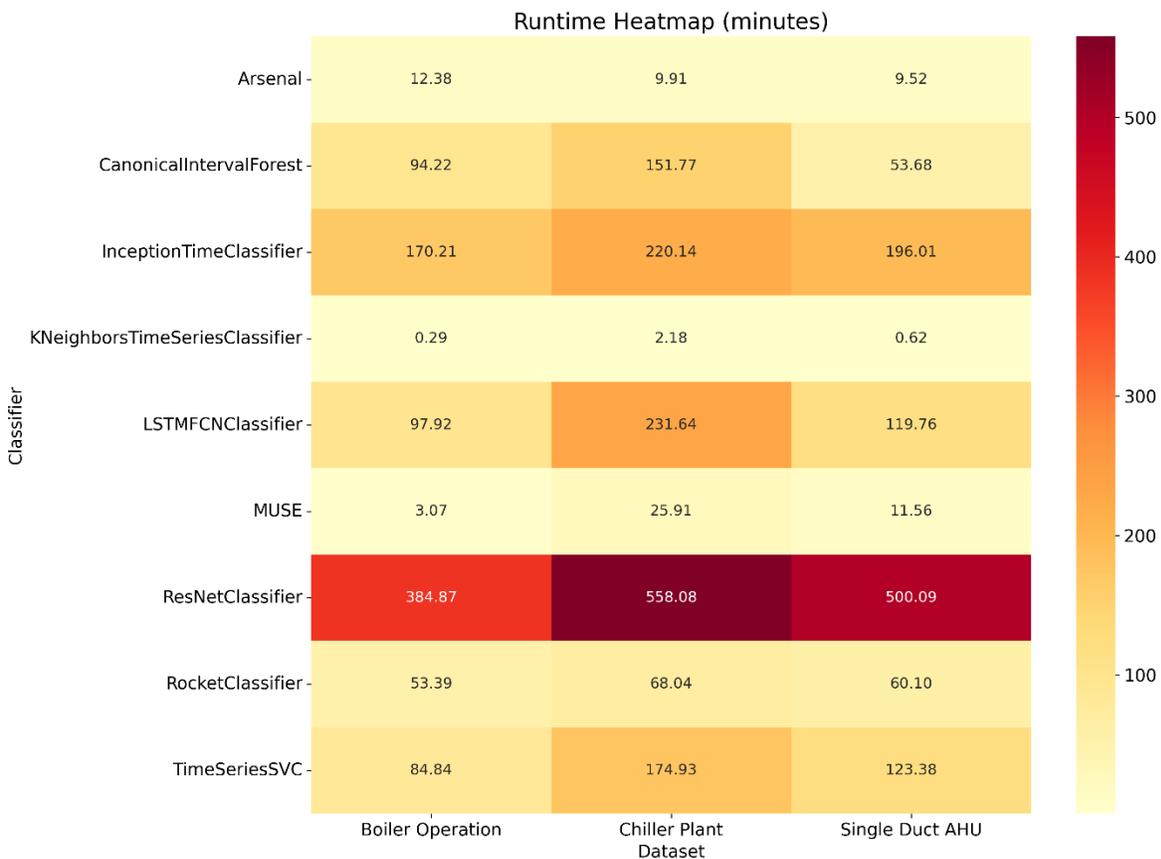


Figure 26: Heat map of runtimes across datasets.

3.3.3 Comparing performance across classifiers categories

The box plot in Figure 27: offers an overview of accuracy distribution across different classifier categories. Deep learning-based and interval-based classifiers demonstrate the highest median accuracy and the largest interquartile range, suggesting they generally perform well but with significant variability across datasets or individual classifiers within the category. Distance-based classifiers show a high median accuracy as well, but with a smaller spread, indicating more consistent performance. In contrast, convolutional-based classifiers exhibit the poorest performance, with the lowest median accuracy and several outliers, pointing to potential limitations or mismatches between these algorithms and the nature of the datasets.

In Figure 28 the average accuracy of the classifiers across datasets vs the average runtime is plotted. This plot offers a comprehensive comparison of various classifier categories, evaluating their performance based on average runtime (displayed on a logarithmic scale) and average accuracy. The data reveals a general trend where increased computational time correlates with higher accuracy, though with notable exceptions. Distance-based methods emerge as the most efficient, boasting the fastest runtime while maintaining respectable accuracy. In contrast, deep learning and interval-based approaches achieve the highest accuracy but at the cost of significantly longer processing times. Interestingly, convolutional-based methods stand out as an anomaly, consuming substantial computational resources yet yielding surprisingly low accuracy.

The Boiler dataset stands out as the one where all classifier categories achieve their highest accuracy, with deep learning-based, distance-based, and interval-based methods all reaching exceptionally high accuracy (0.976 - 0.979). This suggests that the Boiler Operation data may have clearer patterns or be easier to classify overall. In contrast, the Single Duct AHU dataset appears to be the most challenging, with all categories showing their lowest accuracy scores on this dataset, though deep learning-based methods maintain a relative advantage.

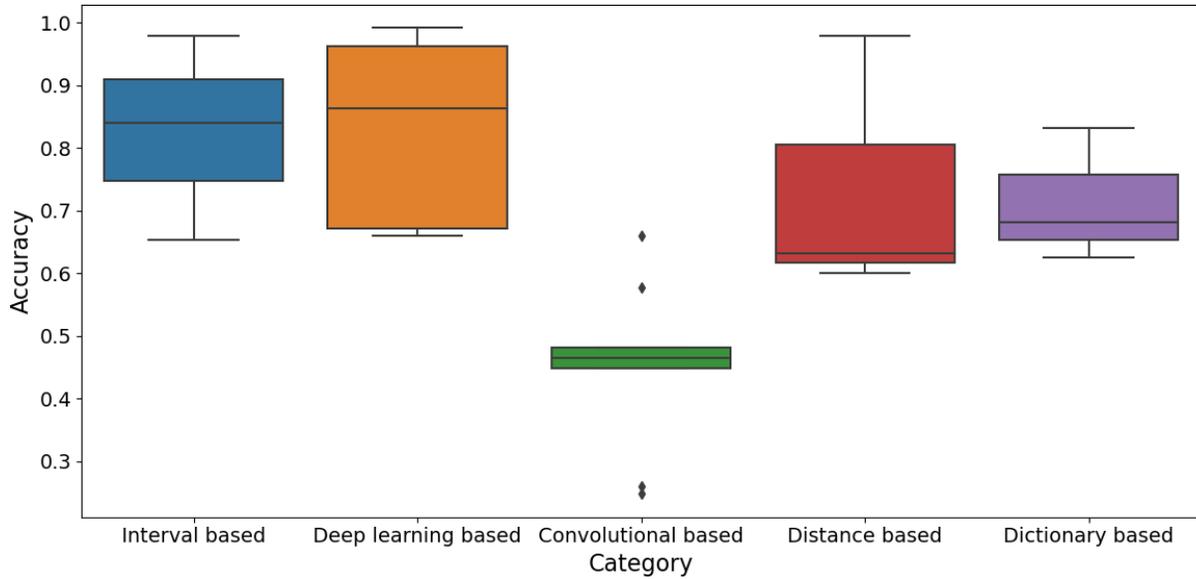


Figure 27: Box plot of the average algorithm's performance across datasets by category.

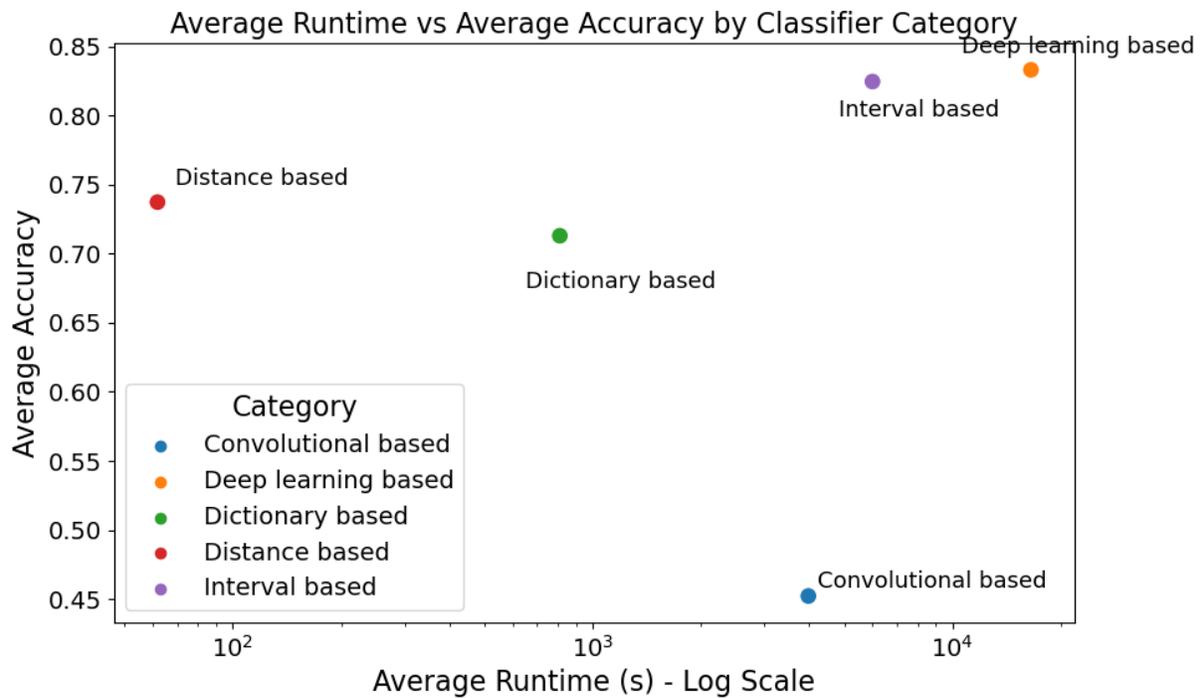


Figure 28: Average accuracy vs runtime per classifier category.

To further understand the performance variations, statistical analyses were conducted on the results. Although CIF and KNN with DTW showed higher average metrics, the differences were

not statistically significant (p-values > 0.05). This analysis suggests that while some classifiers may perform better on average, the differences might not be meaningful in practical applications without further optimization. The potential for improving classification results through advanced feature engineering is considerable. Adjusting data sampling rates and enhancing feature extraction techniques could lead to significant performance gains, as indicated by the initial discrepancies observed in classifiers' performances across the three datasets. Exploring these avenues could provide deeper insights and more robust models for fault detection in HVAC systems.

The extended analysis of classifier performance across multiple datasets highlights the importance of algorithm selection based on the specific characteristics of the data. While CIF and KNN with DTW are generally robust across various settings, their effectiveness can be enhanced through tailored feature engineering. This study's findings underline the need for comprehensive testing and optimization to deploy effective fault detection and diagnostics solutions in real-world scenarios.

3.3.4 Comparison with tree-based ensemble method

While algorithms like InceptionTime and CIF demonstrated good overall performance with average accuracy of 0.8 across the three datasets, it is worth comparing their performance to supervised ensemble methods. In this subsection, the comparison to the method used in chapter 2, XGBoost, is conducted. The results can be seen in Table 12.

Table 12: Performance comparison between XGBoost and LSTMFCNC across datasets

Classifier	SD-AHU					Chiller plant					Boiler plant				
	Runtime (ms)	Accuracy	Precision	Recall	F1 (ms)	Runtime (ms)	Accuracy	Precision	Recall	F1 (ms)	Run time (ms)	Accuracy	Precision	Recall	F1
XGBoost	0.21	0.62	0.62	0.62	0.56	0.16	0.92	0.90	0.92	0.91	0.14	0.99	0.97	0.98	0.99
LSTMFC NC	120	0.66	0.61	0.66	0.62	232	0.87	0.86	0.87	0.85	98	0.98	0.98	0.98	0.98

The results show that overall, XGBoost has similar results to LSTMFCNC but with extremely low runtime. For the single duct AHU, LSTMFCNC scores slightly higher with an accuracy of 0.66 vs

0.62 for the XGBoost. While the runtime for LSTMFCNC is 120 minutes vs 0.21 minutes for XGBoost. For the chiller plant dataset, the XGBoost has a slightly better performance with an accuracy of 0.92 and runtime of 0.16 minutes vs 0.87 accuracy for LSTMFCNC and runtime of 232 minutes.

The similar performance of both models might suggest that while there are temporal patterns in the three datasets, they may not be the dominant factor in distinguishing between classes. This means that spatial dependencies -correlation between features- in the three datasets are as important as temporal dependencies since XGBoost does not explicitly model the temporal dependencies.

4 Self-Supervised Transformer based architecture for fault detection

4.1 Background

Recent research in the building sector has concentrated on two learning approaches to address the scarcity of labelled data: transfer learning and semi-supervised learning [117], [118]. Solutions based on transfer learning propose using insights gleaned from data-rich buildings to tailor models for buildings with less data [119], [120]. This approach offers a promising way to exploit operational data from various building systems and conditions. However, it presumes the availability of data from buildings with similar characteristics, which may not always be the case. In focusing on the data from individual buildings, other studies have assessed the merits of semi-supervised learning in using unlabelled operational data [121]. Yan et al. examined the efficacy of various semi-supervised algorithms in categorizing faults in AHUs [122], finding that this approach can significantly enhance model performance even with limited labelled data. Fan et al. introduced a unique semi-supervised framework using artificial neural networks for diagnosing faults in AHUs, employing a base model trained on limited labelled data and iteratively updating it with high-quality pseudo labels derived from unlabelled data [123]. Li et al. applied semi-supervised generative adversarial networks to better understand the distribution of unlabelled data, thereby improving fault diagnosis in chillers [124], [125]. Their approach involved training a discriminator model to classify real data labels while distinguishing between real and artificial data samples, thus facilitating the creation of a reliable fault classification model with minimal labelled data. A notable limitation of semi-supervised learning is its partial dependence on initial labelled data. For example, in the widely used self-training method, the quality of pseudo labels generated from unlabelled data can be substandard if the initial model is developed with an extreme scarcity

of data, potentially leading to decreased performance in predictive modelling. One of the main reasons behind the lack of adoption of data driven FDD in the building sector is due to the fact that most proposed methods depend entirely or partially on labelled data which is inherently difficult to systematically obtain for several reasons:

1. expertise requirement: accurately labelling faults requires a deep understanding of building systems and operations, which necessitates the involvement of domain experts. This can significantly increase the time and cost associated with the data labelling process;
2. variability and complexity: buildings vary greatly in their design, usage, and maintenance, leading to a wide range of potential faults that are often complex and interrelated. This variability makes it challenging to create a comprehensive labelling schema that accurately represents all potential faults;
3. dynamic environments: the operational conditions of buildings and their systems can change over time, affecting fault manifestations. This dynamic nature requires continuous updates to labelled data to remain relevant, adding to the complexity and cost of the labelling process.

Self-supervised learning emerges as a promising solution, offering a potential means to reduce the reliance on labelled data in predictive modelling [126] Self-Supervised Learning (SSL) represents a segment of unsupervised learning that leverages internally generated tasks, known as pretext tasks, to extract supervisory cues from data without labels. These internally devised challenges enable the model to extract knowledge from the dataset, which in turn fosters the creation of meaningful representations for subsequent analytical tasks. SSL circumvents the need for externally labelled data since the supervisory signals are intrinsically obtained from the data. Owing to the strategic design of these pretext tasks, SSL has marked notable advancements in the realms of Computer Vision (CV) and Natural Language Processing (NLP).

In this chapter, we used and trained an encoder-only transformer-based architecture in a generative, self-supervised manner. This method was tested against unlabelled data from a real building equipped with a heat pump (HP) that is connected to an AHU for ventilation and a floor heating system.

4.2 Transformer introduction & applications

The goal of this section is to give an overview of the original transformer architecture as it was introduced for Natural language processing (NLP) purposes and all its components (positional encoding, multi head attention, feed forward and residual network). Then how this architecture was adapted to be used for time series data.

4.2.1 Vanilla transformer

The innovation of Transformer in deep learning [127] has brought great interest recently due to its excellent performances in NLP [128] computer vision (CV) [129], and speech processing [130]. Over the past few years, numerous Transformer variants have been proposed to advance the state-of-the-art performances of various tasks significantly. There are quite a few literature reviews from different aspects, such as in NLP applications [131], CV applications [132], and efficient Transformers [133].

The classic Transformer introduced by [127] is essentially built on an encoder-decoder framework. This structure comprises multiple identical layers in both the encoder and decoder. Each layer is characterized by two main components: a multi-head attention mechanism and a position-specific feed-forward network. The decoder further integrates a cross-attention mechanism that works in tandem with the multi-head self-attention and the position-wise feed-forward module.

4.2.1.1 Encoding the input and position

In contrast to models like LSTM and RNN, the basic Transformer doesn't use a recurrent mechanism. Instead, it adds positional encoding to the input embeddings to capture sequential information. We briefly explain some prominent positional encoding methods. In the standard Transformer, each sequence position, denoted as:

$$PE(t)_i = \begin{cases} \sin(\omega_i t) & i\%2 = 0 \\ \cos(\omega_i t) & i\%2 = 1 \end{cases} \quad (9)$$

$\omega_i t$ represents a predefined frequency for each dimension. An alternative approach is to learn these positional embeddings, which offers more adaptability, as suggested by [134]. For relative positional encoding the is that the relationships between sequence positions can be more

informative than their absolute positions. Some techniques have been devised to add relative positional encodings directly to the attention mechanism's keys. Shaw and team in 2018 provided insights into this. Additionally, there are hybrid methods that merge both absolute and relative positional encodings, where the positional information gets combined with the token embeddings directly.

4.2.1.2 Multi-head attention

With Query-Key-Value (QKV) model, the scaled dot-product attention used by Transformer is given by:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}}\right)\mathbf{V} \quad (10)$$

where queries $\mathbf{Q} \in \mathcal{R}^{N \times D_k}$, keys $\mathbf{K} \in \mathcal{R}^{M \times D_k}$, values $\mathbf{V} \in \mathcal{R}^{M \times D_v}$, N , M denote the lengths of queries and keys (or values), and D_k , D_v denote the dimensions of keys (or queries) and values. Transformer uses multi-head attention with H different sets of learned projections instead of a single attention function as:

$$\text{MultiHeadAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)\mathbf{W}^O \quad (11)$$

where $\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$. The $\text{Attention}()$ function computes the relevance of different values based on the queries and keys. For each query, $\text{Attention}()$ assigns weights to the keys based on their similarity, and these weights are used to aggregate the corresponding values into a single output. This allows the model to focus on the most relevant parts of the input when making predictions.

4.2.1.3 Feed-forward and Residual Network

In this formula \mathbf{H}' represents the output from the preceding layer. \mathbf{W}^1 is a matrix of dimensions appropriate for mapping the input features to an intermediary dimension, while \mathbf{W}^2 serves to map these intermediary features to the desired output dimension. Similarly, \mathbf{b}^1 and \mathbf{b}^2 are bias vectors corresponding to each weight matrix and are subject to optimization during training.

$$FFN(\mathbf{H}') = \text{ReLU}(\mathbf{H}'\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}^2, \quad (12)$$

In this formula \mathbf{H}' represents the output from the preceding layer. \mathbf{W}^1 is a matrix of dimensions appropriate for mapping the input features to an intermediary dimension, while \mathbf{W}^2 serves to map these intermediary features to the desired output dimension. Similarly, \mathbf{b}^1 and \mathbf{b}^2 are bias vectors corresponding to each weight matrix and are subject to optimization during training. As the network depth increases, it becomes beneficial to incorporate a residual connection, along with layer normalization, to enhance the flow of gradients during training. Thus, the module can be extended as follows:

$$\begin{aligned} \bar{H}' &= \text{LayerNorm}(\text{SelfAttn}(X) + X) \\ H &= \text{LayerNorm}(FCN(\bar{H}') + \bar{H}') \end{aligned} \quad (13)$$

Here $\text{SelfAttn}()$ signifies the self-attention mechanism that processes the input X . and $\text{LayerNorm}()$ denotes the process of layer normalization.

4.2.2 Transformers for time series and anomaly detection

In recent advancements, the Transformer architecture, originally designed for natural language processing, has been extensively modified to cater to the intricacies of time series data [135], [136]. One pivotal adaptation is the introduction of adaptive positional encoding techniques, moving beyond the vanilla model's basic positional encoding. Research indicates dynamic embeddings derived directly from time series data, such as those introduced by learning layers within the Transformer [137] or generated through LSTM networks [138], significantly enhance model efficacy by providing tailored flexibility and capturing the sequential order inherent in time series.

Moreover, leveraging timestamps as an additional form of positional encoding, as seen in models like Informer [139], Autoformer [140], and FEDformer [141], brings forth the untapped potential of time-specific data points. This approach underscores the value of incorporating both regular intervals and significant dates to enrich the model's temporal understanding.

Addressing the computational challenges of the self-attention mechanism, proposals like LogTrans [142] and Pyraformer [143] have introduced efficient strategies through inducing sparsity and

exploiting the self-attention matrix's low-rank characteristics, respectively. Architectural innovations further include hierarchical structuring, as implemented by Informer [139] and Pyraformer [143], to process time series at varying scales, enhancing both model efficiency and data interpretation capabilities.

Transitioning to anomaly detection, the transformative application of the Transformer architecture [144] and its integration with generative neural models such as VAEs [145], [146], [147], [148] and GANs [149] have marked significant improvements in detecting time series anomalies. Adversarial training methods [147], multi-scale approaches [146], and graph-based learning frameworks exemplify the broadening scope of Transformers in capturing complex temporal relationships and multivariate series characteristics. These adaptations underscore the architecture's versatility in enhancing anomaly detection accuracy and addressing the limitations of traditional methods.

In sum, these modifications, and applications of the Transformer architecture to time series analysis and anomaly detection highlight the ongoing innovation in adapting deep learning models to the unique demands of time series data, significantly improving their performance and applicability across various tasks.

4.3 Methodology of fault detection using a Transformer architecture

In this section we introduce the data preprocessing procedure, the model used in the study, the self-supervised training method used for the pretraining step and finally the dynamic thresholding technique used to flag the anomalies.

4.3.1 Core Architecture

Central to our approach is an encoding mechanism inspired by the transformer architecture delineated by [127]. Our model diverges from this foundational design in that it eschews the decoder module, opting instead for an encoder-only framework. The primary reason for employing only an encoder in this research, focusing on multivariate anomaly detection in time series, is due to the non-generative nature of the task. Unlike the original Transformer architecture, which was designed for language translation - a generative task requiring an encoder to understand one

language and a decoder to generate another - anomaly detection in time series data involves identifying deviations from normal patterns within the same data context. Therefore, a decoder is unnecessary; the encoder alone is sufficient to model and identify these anomalies effectively. This approach streamlines the model and makes it more computationally efficient, focusing its learning capabilities on recognizing irregularities in the time series data. The computational efficiency improvement stems from the fact that in Central to our approach is an encoding mechanism inspired by the transformer architecture delineated by [127]. Our model diverges from this foundational design in that it eschews the decoder module, opting instead for an encoder-only framework. The primary reason for employing only an encoder in this research, focusing on multivariate anomaly detection in time series, is due to the non-generative nature of the task. Unlike the original Transformer architecture, which was designed for language translation - a generative task requiring an encoder to understand one language and a decoder to generate another - anomaly detection in time series data involves identifying deviations from normal patterns within the same data context. Therefore, a decoder is unnecessary; the encoder alone is sufficient to model and identify these anomalies effectively. This approach streamlines the model and makes it more computationally efficient, focusing its learning capabilities on recognizing irregularities in the time series data. The computational efficiency improvement stems from the fact that in traditional encoder-decoder architectures, both the encoder and decoder independently contribute to computational complexity due to the self-attention mechanism's pairwise comparison of tokens, resulting in a quadratic relationship with the input sequence length. By adopting an encoder-only model, we remove the need for the decoder and its associated complexity entirely. In the context of anomaly detection, where the decoder's generative function is not required, our approach effectively halves the self-attention computation. Therefore, for a time series of length (n), while an encoder-decoder model would require $O(2.l.n^2)$ operations for l operations due to the combined processing in both the encoder and decoder, our encoder-only model require only $O(l.n^2)$ operations. This is a conservative estimate, as it does not factor in the additional computational load imposed by the autoregressive nature of the decoder, which cannot be parallelized across sequence positions. We provide an illustrative representation of our model's universal structure in Figure 29, applicable to an array of tasks. Table 13 provides a detailed breakdown of each layer within our model, its functionality, and the extent of its utilization within the core architecture. The reader is directed to the seminal transformer literature for a

comprehensive elucidation of the model, whilst this discourse will focus on the modifications, we introduced to facilitate the processing of multivariate temporal sequences as opposed to linguistic token sequences.

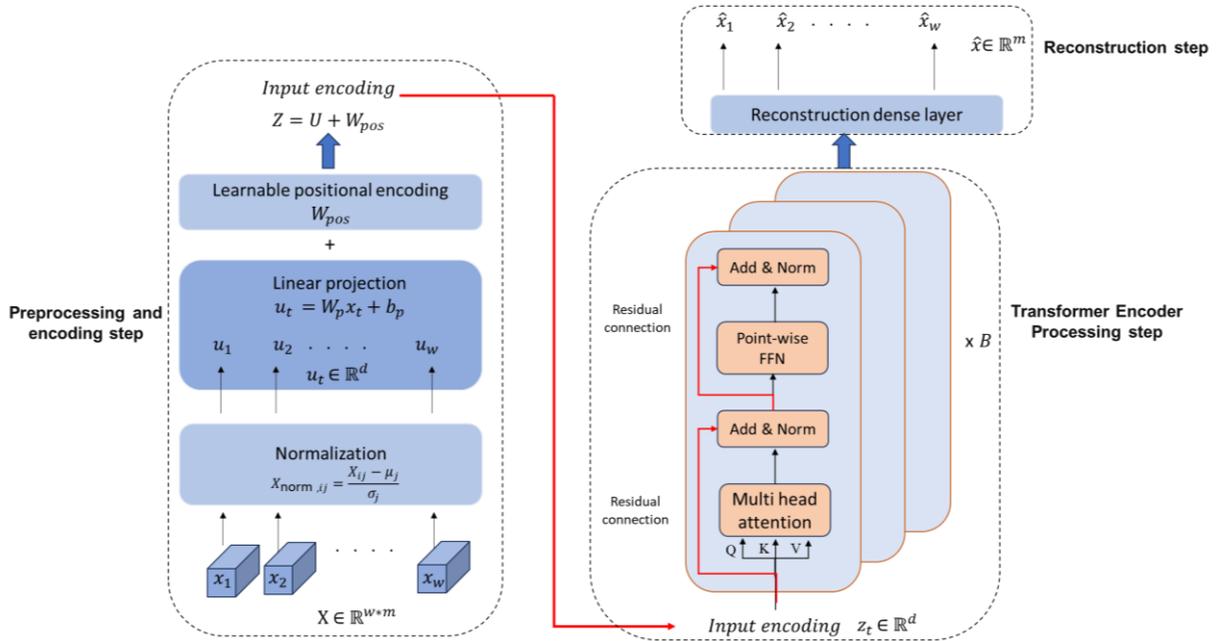


Figure 29: on the left figure (starting from the bottom): the main architecture of the algorithm the steps of preprocessing and encoding of the data to the model dimension (d) are displayed; on the right the modelling and reconstruction of the data to the origin.

Table 13: Model architecture breakdown.

Layer name	Layer type	Description	Number of layers
Input normalization	Preprocessing	Standardizes the input data to zero mean and unit variance.	1 per feature
Linear projection	Transformation	Projects normalized features into a d -dimensional model space.	1
Positional encoding	Encoding	Adds learnable temporal context to input sequences.	1
Multi head attention	Self-Attention	Processes sequences in parallel, focusing on different parts of the sequence simultaneously.	2 (7 attention heads each)
Feedforward network	Transformation	Applies point-wise transformations to the output of the attention layer.	2
Add & Norm	Residual connection	Combines the outputs of the attention and feedforward networks with layer normalization.	4 (2 per encoder layer)
Output projection	Reconstruction	Maps the encoded sequence back to the original feature space for reconstruction.	1

Each datum for training, denoted as X within the real value space \mathbb{R}^{w*m} , represents a multivariate temporal sequence comprising w instances across m distinct variables, thus forming a series of feature vectors x_t within \mathbb{R}^m . Prior to dimensionality transformation, the feature vectors x_t are subjected to a normalization process—subtracting the mean and scaling by the variance computed across the training dataset—and subsequently projected linearly into a d dimensional vector space, d being the inherent dimensionality of the transformer's internal sequence representation, often referred to as the model dimension:

$$u_t = W_p x_t + b_p \quad (14)$$

Herein $W_p \in \mathbb{R}^{d*m}$ and $b_p \in \mathbb{R}^d$ are parameter matrices and vectors subject to optimization, with $u_t \in \mathbb{R}^d$ representing the series of model inputs analogous to the lexical embeddings in linguistic transformers. These inputs are subsequently transformed into the queries, keys, and values for the self-attention mechanism upon integration of positional encodings and subsequent application of the associated transformation matrices. The transformer, inherently a feed-forward construct, lacks innate sensitivity to input sequence order. To instill an awareness of temporal structure within the model, we introduce positional encodings $W_{pos} \in \mathbb{R}^{w*d}$ into the input vector sequence $U \in \mathbb{R}^{w*d} = [u_1, u_2 \dots u_w]$, thereby obtaining $Z = U + W_{pos}$.

In a departure from the fixed, sinusoidal positional encodings posited in the original transformer paradigm, our model utilizes a set of positional encodings that are subject to optimization. This alteration is substantiated by empirical evidence indicating enhanced performance across all considered datasets. These learnable encodings seem to minimally interfere with the temporal data's quantitative attributes. We postulate that this is attributable to the encodings evolving to occupy a vector subspace that is approximately orthogonal to that of the time series data, a hypothesis supported by the higher-dimensional nature of the embedding space which simplifies the attainment of orthogonality. In this study, time2vec method [150] was used to encode time stamps in the data. In a departure from the fixed, sinusoidal positional encodings posited in the original transformer paradigm, our model utilizes a set of positional encodings that are subject to optimization. This alteration is substantiated by empirical evidence indicating enhanced performance across all considered datasets. These learnable encodings seem to minimally interfere

with the temporal data's quantitative attributes. We postulate that this is attributable to the encodings evolving to occupy a vector subspace that is approximately orthogonal to that of the time series data, a hypothesis supported by the higher-dimensional nature of the embedding space which simplifies the attainment of orthogonality. In this study, time2vec method [150] was used to encode time stamps in the data.

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i\tau + \varphi_i, & \text{if } i = 0 \\ \mathcal{F}(\omega_i\tau + \varphi_i), & \text{if } 1 \leq i \leq k \end{cases} \quad (15)$$

Where $\mathbf{t2v}(\tau)[i]$ is the i^{th} element of $\mathbf{t2v}(\tau)$. \mathcal{F} is a periodic activation function and ω_i s and φ_i s are learnable parameters. Time series data is inherently variable in length. Our architecture effectively addresses this heterogeneity by establishing a uniform maximum sequence length w for the dataset. Sequences falling short of this length are augmented. The model was trained on a window size of 96 corresponding to one day of measurement. We used 2 layers of transformer encoders and 2 layers of feed forward unit of encoders. While 7 heads were used in the multi head attention.

4.3.2 Self-supervised learning pre-training

For the foundational self-supervised pre-training phase of our model, we engage an autoregressive task wherein a portion of the input data is occluded with zeros, compelling the model to predict the concealed information. This process entails the systematic obscuration of subsets of the input sequence—achieved through the multiplication of the input $X \in \mathbb{R}^{w \times m}$ with binary mask M , generated independently for each sample. In this masking schema, a proportion r of each mask column (equivalent to a singular variable in the time series) oscillates between segments of zeros and ones, following a predetermined state transition probability distribution to determine the length of each obfuscated segment, thereby generating sequences with a geometric distribution characterized by a mean unmasked segment length l_u and a mean masked segment length l_m , as given by $l_u = \frac{1-r}{r} l_m$ with l_m being set to 3 for the conducted experiments.

We adopt this masking strategy—distinct from the "cloze" method employed in NLP models such as BERT—where the masked values in the time series are supplanted by zeros, as opposed to replacing word embeddings. This method is designed to incite the model to not only predict the

immediate succeeding values but also to integrate the temporal dependencies between variables. A linear layer with optimizable parameters $W_o \in \mathbb{R}^{m \times d}$, $b_o \in \mathbb{R}^m$ is applied to the terminal vector representations $z_t \in \mathbb{R}^d$ at each time step, with the model simultaneously estimating the complete unobscured input vectors x_t ; however, the Mean Squared Error (MSE) is computed solely for the predictions on the masked segments as indicated by the mask set $M = \{t_i: m_{ti} = 0\}$ where m_{ti} are the elements of the mask M . The MSE for each data sample is as follows:

$$\begin{aligned} \tilde{x}_t &= W_o z_t + b_o & (16) \\ L_{MSE} &= \frac{1}{|M|} \sum_{(t,i) \in M} (\tilde{x}(t,i) - x(t,i))^2 \end{aligned}$$

This pre-training objective is methodologically divergent from denoising autoencoders as it does not consider the entire input reconstruction but rather focuses on the masked segments. Notably, this approach is not reliant on assumptions of noise characteristics typically postulated in denoising paradigms, such as Gaussian distributions. The design also takes into account the distributions of the actual masked values and the subsequent impact on learning. The pre-training step overview can be seen in Figure 30.

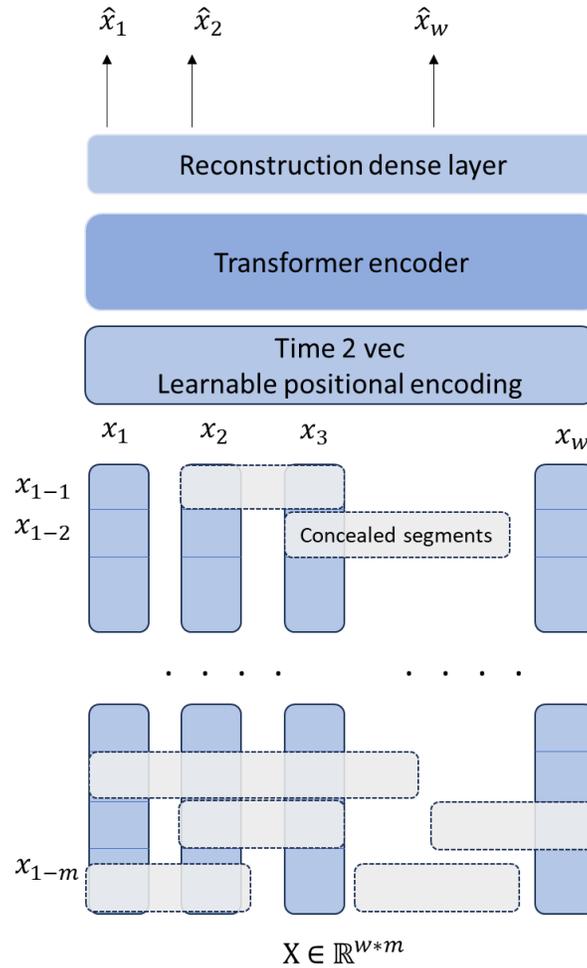


Figure 30 Pre-training step overview.

4.3.3 Dynamic thresholding and fine tuning

After the reconstruction of the multivariate time series, the anomaly scores are calculated using the absolute difference between the original and the predicted ones, multiplied by the average attention weights of each window averaged over multiple heads. Dynamic thresholding technique is then applied to the anomaly scores to flag anomalies that exceed the threshold.

In this work, I implement the Peak Over Threshold (POT) method, which enables the automatic and dynamic selection of thresholds [151] and used by [152]. This technique is grounded in the principles of extreme value theory, facilitating the fitting of data distributions using a Generalized Pareto Distribution. The first step in the POT method is to set an initial threshold t in a window w . This threshold is set such that only the most extreme values in the data set for each window are considered for further analysis. In this research this value was chosen as 85th percentile with a

window of 2 hours. Once the threshold is set, the method focuses on the excesses over this threshold. These excesses are defined as:

$$Y_i = X_i - t \quad (17)$$

Where Y_i is the excess over the threshold, X_i is the individual anomaly score at a certain time stamp and t is the initial threshold. The distribution of excesses over the threshold is fitted to generalized pareto distribution (GPD). The GPD is characterized by two parameters: scale parameter σ and the shape parameter γ . The cumulative distribution function (CDF) of GPD is given by:

$$G(y; \sigma, \gamma) = \begin{cases} 1 - \left(1 + \frac{\gamma y}{\sigma}\right)^{-\frac{1}{\gamma}}, & \text{if } \gamma \neq 0 \\ 1 - \exp\left(-\frac{y}{\sigma}\right), & \text{if } \gamma = 0 \end{cases} \quad (18)$$

where $y > 0, \sigma > 0$, and $y \leq -\frac{\sigma}{\gamma}$ for $\gamma < 0$. The σ, γ are the scale and the shape parameters respectively. Those parameters are estimated using the Maximum Likelihood Estimation (MLE). In this paper we used the Grimshaw tricks [49] to calculate the maximum value of maximum likelihood function.

Once the GPD parameters are estimated, the distribution in each window can be used to assess the extremeness of new observations. An observation is flagged as an anomaly if its excess over the threshold has a low probability under the estimated GPD. This is typically done by computing the quantile or the survival function of the GPD for a new observation and comparing it to a pre-defined risk level q . If the probability of observing an excess over the threshold is lower than q .

Finally, the quantiles are calculated for a given probability level using the inverse of the GPD's CDF. This quantile represents the value for which there is a probability q that the observed value will exceed it. The formula for quantile calculation under GPD is:

$$z_q = t + \frac{\sigma}{\gamma} ((1 - q)^{-\gamma} - 1) \text{ for } \gamma \neq 0 \quad (19)$$

Where z_q is the quantile for a probability of q . The calculated quantiles will be used as the calculated threshold for the anomaly scores. In the case of $\gamma = 0$, the quantile is calculated using the exponential distribution formula. This dynamic threshold is set for each feature's anomaly score. An anomaly is flagged if the anomaly score of any feature exceeded the threshold.

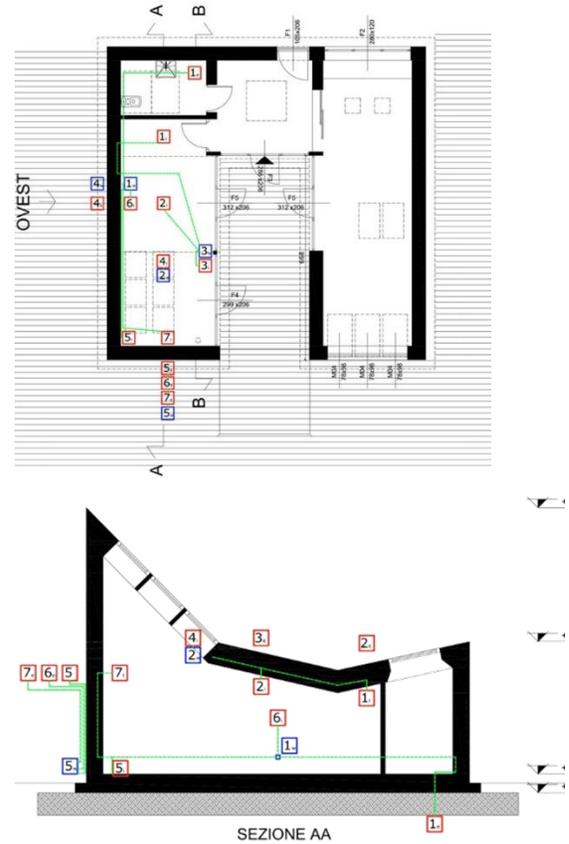
Lastly, if labelled data becomes available, the model can be fine-tuned to further refine its fault detection capabilities. During fine-tuning, the labelled data is used to adjust the encoder's weights through backpropagation, specifically training the model to better classify instances as normal or faulty. This step enhances the model's precision in identifying faults by leveraging direct feedback from the labelled examples.

4.4 Case study

4.4.1 Building envelope and HVAC system

VELUXlab stands as the pioneering Nearly Zero Energy Building in Italy, situated within the confines of a university campus. The journey of VELUXlab began in 2011 when VELUX embarked on a project to transform the Atika demo-house into an innovative laboratory under the auspices of Politecnico di Milano. Initially designed to exemplify a model home suitable for the Mediterranean climate, the building underwent significant enhancements under the guidance of Politecnico di Milano's design team. These upgrades transformed it into an active prototype, offering a tangible example for the development of future sustainable buildings [154].

The retrofit process of the building involved both the improvement of the envelope's layering with new and high performances materials that increased the technical performances of the building case (U-values down to $0,124 \text{ W/m}^2/\text{K}$), and the implementation of systems. Static and dynamic simulations helped to calibrate the design choices to lead through the minimization of energy needs.



(a)

(b)

Figure 31: a) A picture of the building after renovation and localization at Politecnico di Milano, Bovisa Campus; b) plan and a section of the building.

The HVAC system is comprised of air water heat pump as a generation source in the system with 7 kW in heating and 6.1 kW in cooling. As a mechanical ventilation and emission system, air handling unit with maximum flow rate of 470 m³/h with over 90% heat recovery. Radiant floor is also used as an emission system with capacity of 90 W/m² for heating and 30 W/m² for cooling. 11 m² of photovoltaic panels are used, the field is capable of generating 2 kWp.

The HVAC system undergoes continuous monitoring to evaluate the efficiency of its components and to optimize system control, thereby ensuring optimal indoor comfort. This monitoring framework incorporates a range of sensors, including those for temperature, relative humidity, and CO₂, as well as heat and electrical meters. Figure 32 presents a schematic representation of the HVAC system, highlighting the specific locations of these sensors.

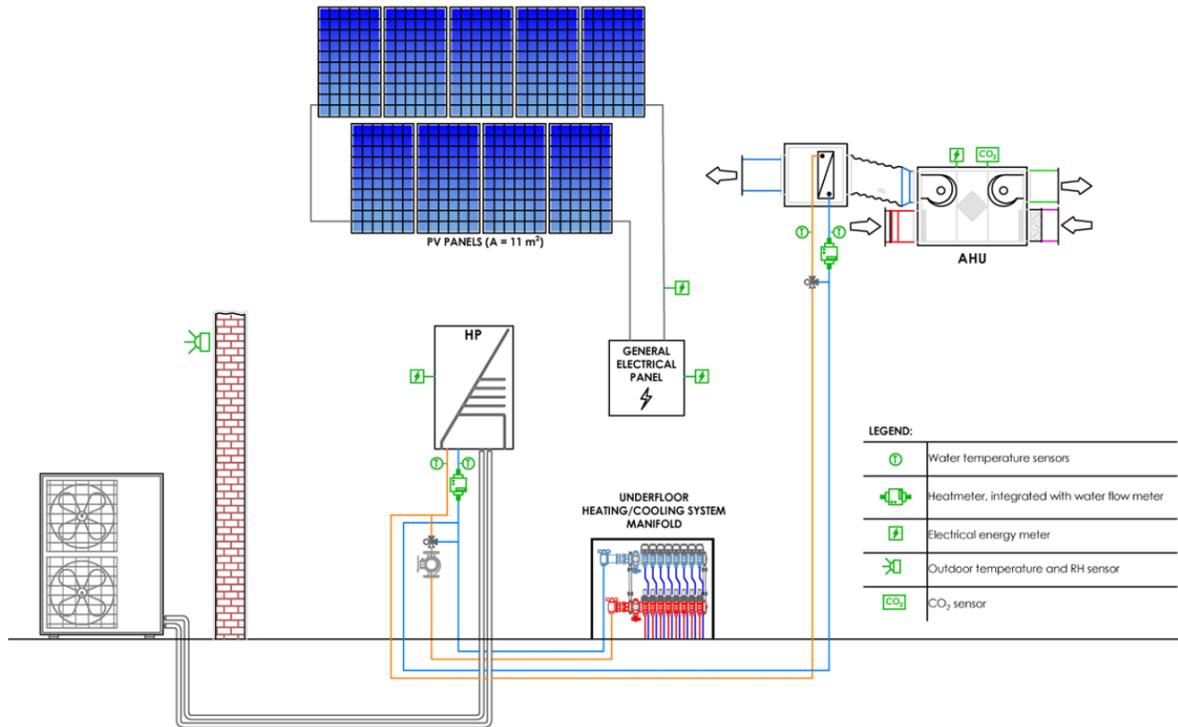


Figure 32 Schematic of the HVAC system and the positions of the sensors of the monitoring system.

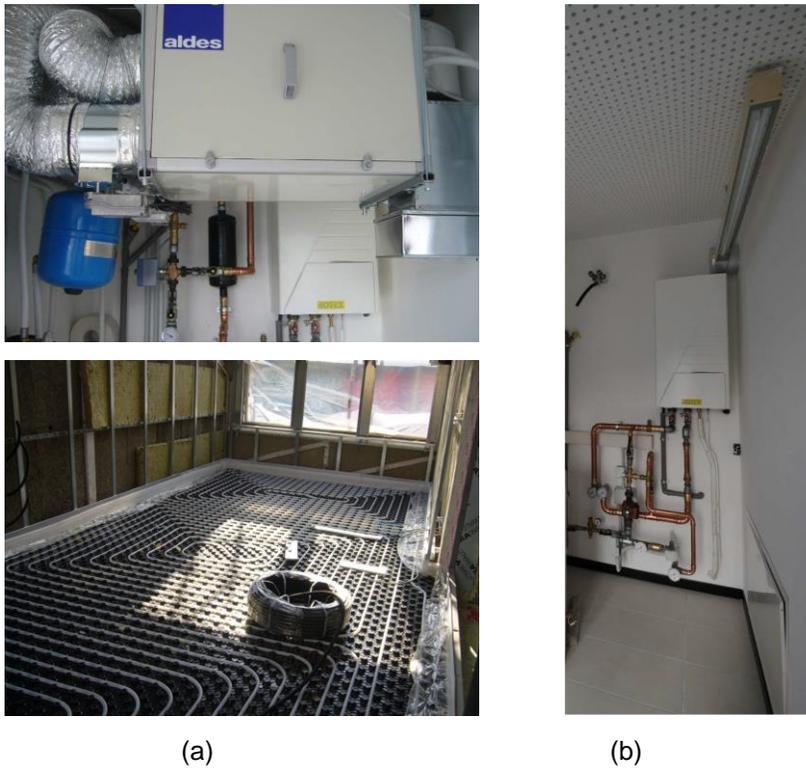


Figure 33: actual system implementation. a): top: Air handling unit; bottom: radiant floor; b): indoor unit of the heat pump.

4.4.2 Data description

The monitoring system initially incorporated over 50 sensors, measuring both numerical and discrete variables. To ensure data integrity, sensors exhibiting more than 15% missing values were excluded, effectively narrowing down the dataset to the most complete time series. The missing values were the results of sensors not logging the values either due to communication errors or sensors malfunction. Further refinement was achieved by evaluating feature correlations; in instances where feature pairs demonstrated a correlation exceeding 95%, one feature from each pair was removed to reduce computational complexity. The data has a range of time steps from 1-10 minutes then later resampled to 15 minutes. The resampling was done by mean aggregation to preserve central tendency of the time series. Analysing the frequency content and probability distribution of the data post-resampling, we confirmed that the mean aggregation process did not introduce significant artifacts or biases. Figure 34 shows the correlation matrix among the features. Correlated features such as AHU damper signal, AHU CO2 control state and AHU fan signal were detected and chosen from. Moreover, features were representing the same measurement but from different monitoring systems -such as external weather station outdoor temperature reading and internal monitoring system reading for external temperature were detected. This process resulted in the selection of 14 features for subsequent analysis, as detailed in Table 14. In Table 15, the uncertainty for each type of measurement is displayed.

Table 14 The measurements from the system that is used in the training of the model.

System component	Measurement
Heat pump	Forward temperature Inlet temperature set point Return temperature Electric power consumption Water flow rate Modulating signal of the mixing valve
Air handling unit	Forward temperature Return temperature. Electric power consumption Water flow rate
Outdoor conditions	Dry bulb temperature Relative humidity
Indoor conditions	Dry bulb temperature CO2 concentration

Table 15 Measurement uncertainty for used features.

Measurement Uncertainty	
Electric power	$\pm 1\%$
Water temperature	$\pm 0.12\text{ }^{\circ}\text{C}$
Temperature	$\pm 0.5\text{ }^{\circ}\text{C}$
Water flow rate	$\pm 2\%$
Relative humidity	$\pm 2\%$
CO ² concentration	$\pm 50\text{ ppm}$

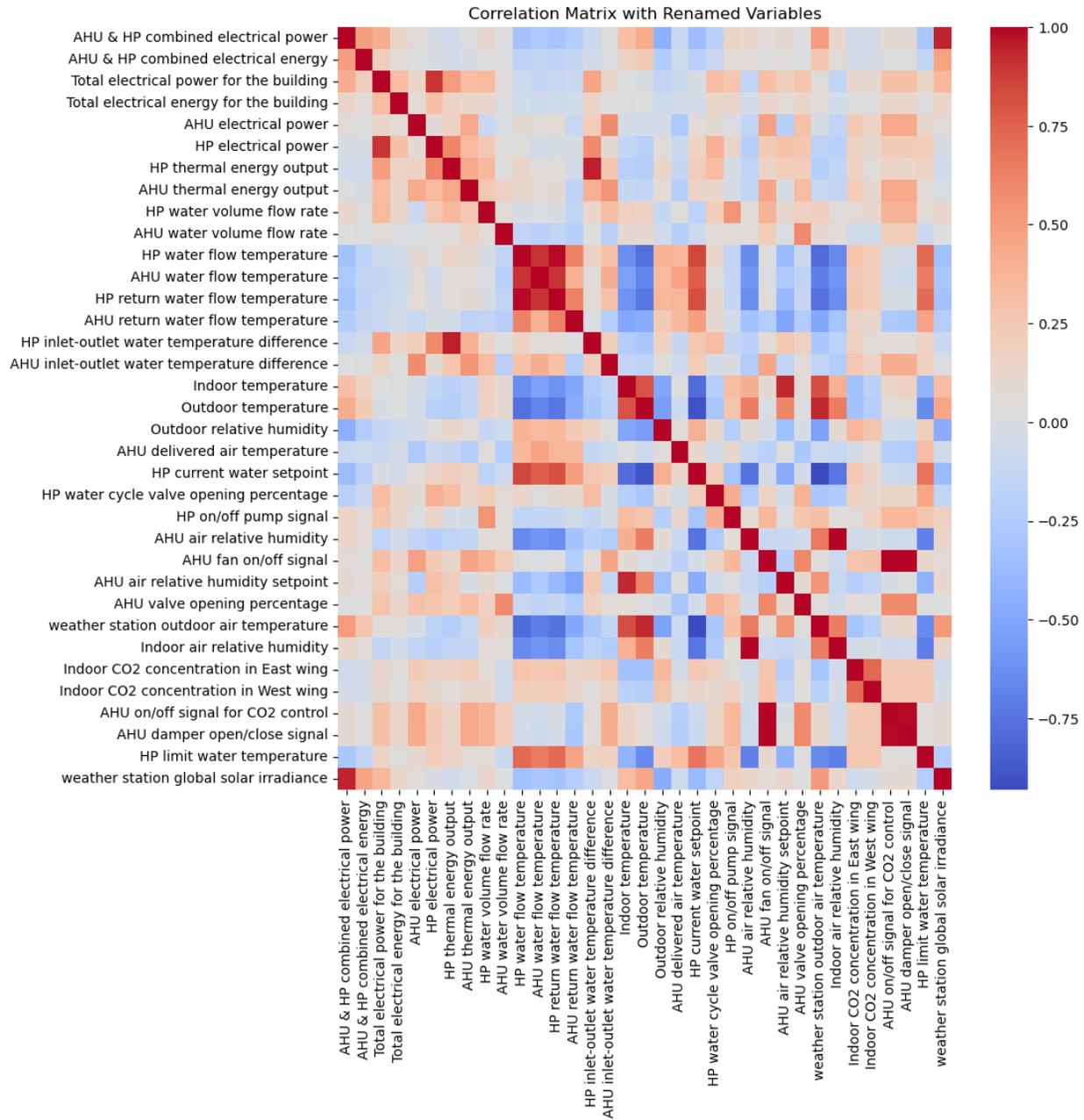
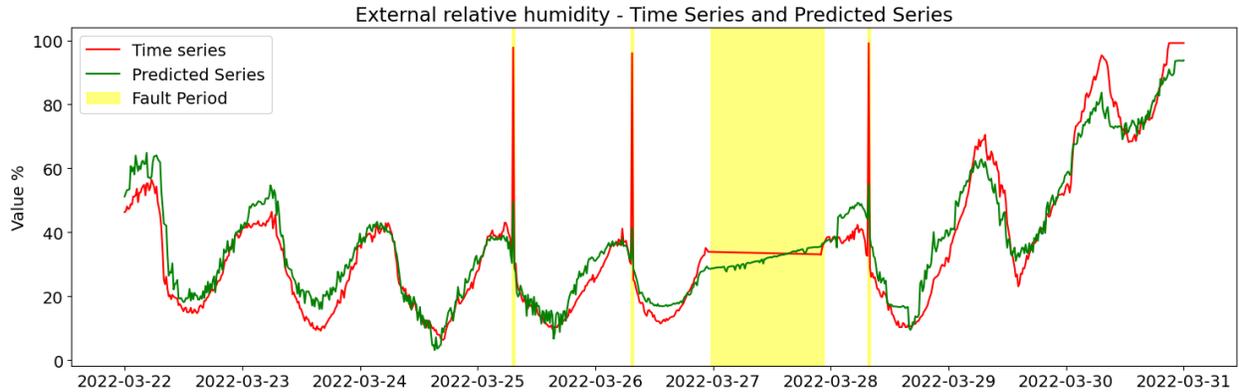


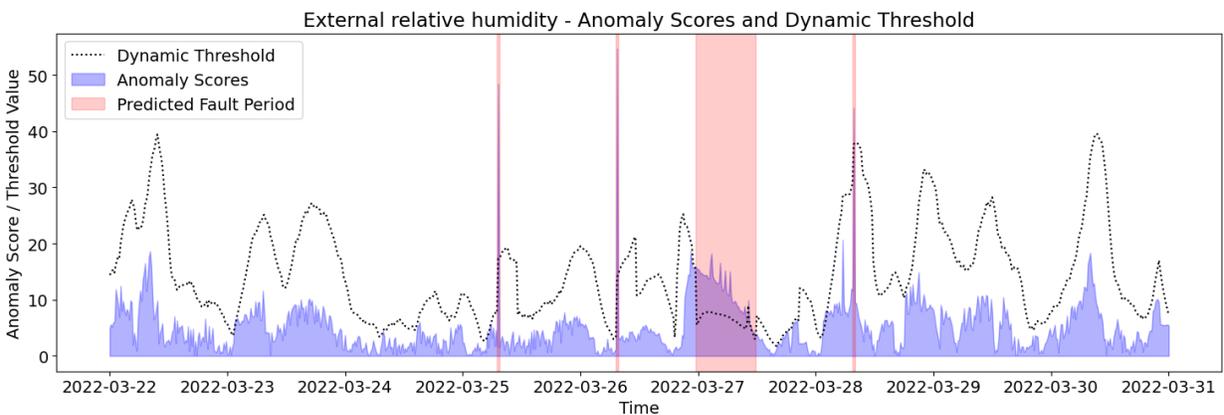
Figure 34 Correlation between the features.

4.5 Results and discussion

As explained previously, anomalous periods are flagged subsequent to the computation of anomaly scores for each feature, upon the application of Peak-Over-Threshold (POT) thresholding. A timestep is classified as anomalous if it surpasses the threshold for any feature.



a) Relative humidity sensor readings, the reconstructed time series from the algorithm.



b) Anomaly scores and the dynamic threshold from actual readings and the reconstruction and the attention scores.

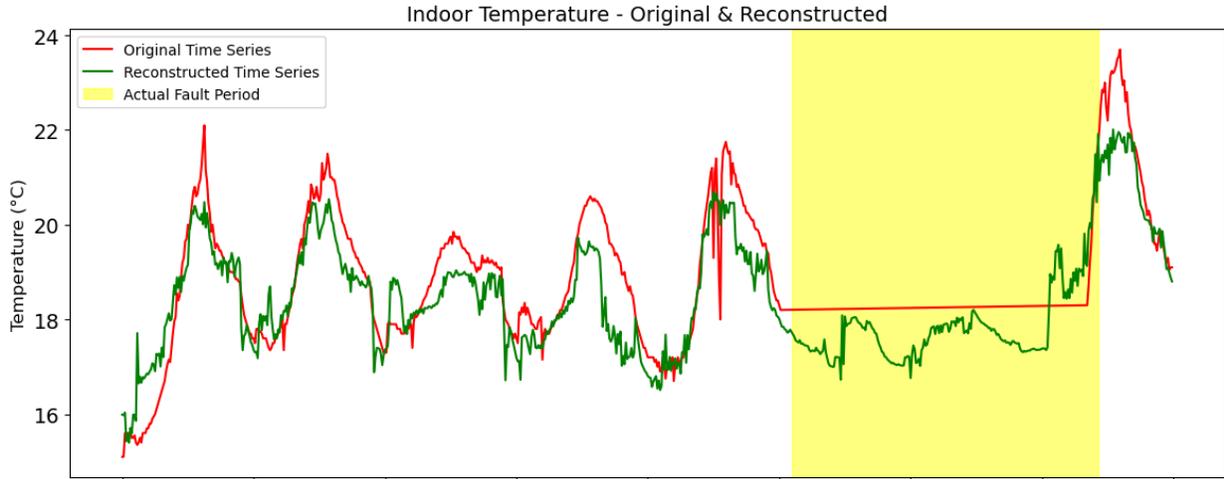
Figure 35: Demonstration of the fault detection process in case of both sequential and point anomalies.

Since no labelled data is available, the assessment of the method was done through two metrics. First how well the reconstructed time series matches the original in the features and the second is analysing the flagged instances. In general, for time series fault instances, there are two types of anomalies, point and sequential. The point anomalies are mostly labelled correctly in the data as demonstrated in Figure 35 where multiple point anomalies in the relative humidity sensor readings are correctly labelled. Those anomalies are quite common in sensor readings and easy to detect for most anomaly detection algorithms. Sequential anomalies on the other hand are much harder to detect and label by anomaly detection algorithms, since it requires identifying the underlying trend or multiple trends in the data and detect the deviation from it. In Figure 36 and Figure 37, sequential anomalies are apparent in the data as the cyclical nature of the trend stops and a non-zero linear trend starts for a period of time. The method proposed was able to detect the change in the trend

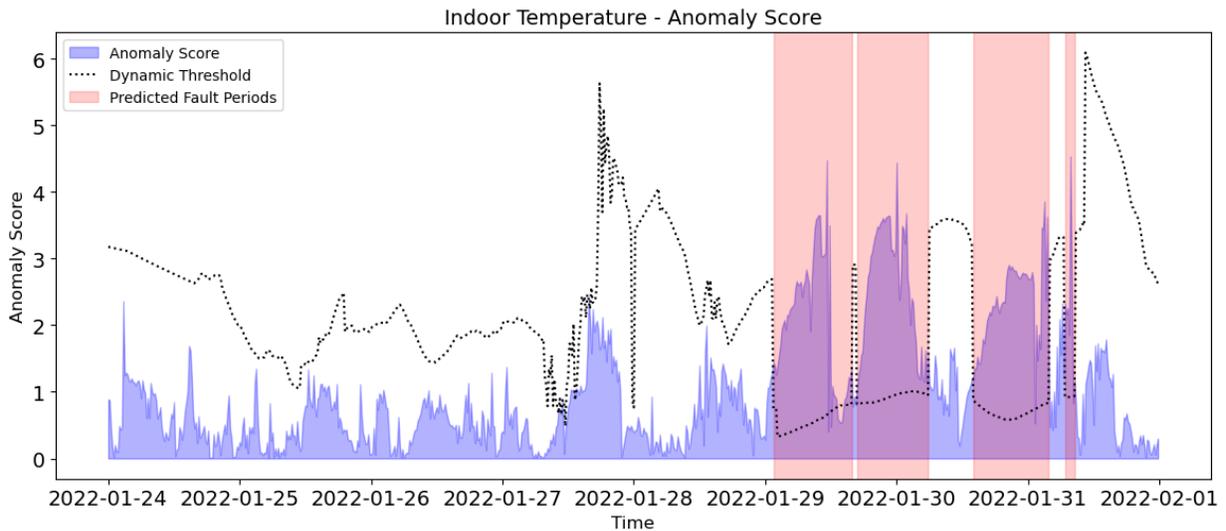
correctly but not for the entirety of the anomalous period. Since the method is primarily built on the anomaly score, which is a function of the reconstruction error, when the reconstruction error reach zero on the points where there is intersection between the reconstructed time series and the original one, the anomaly score reaches zero preventing the continuation of the detection.

In Figure 35, we can see that while the beginning of the fault correctly detected with a spike of anomaly score, the detection stopped when the reconstruction error reached zero and 3 hours of the fault was not detected. In Figure 36, the same behaviour appeared in the indoor sensor readings. Four different anomalous periods were detected and a total of 6 hours out of 50 hours were detected.

The highlighted intervals within Figure 36 are instances where the anomaly score exceeded the designated threshold, signalling a fault.



a) Indoor temperature sensor readings, the reconstructed time series from the algorithm



b) Anomaly scores and the dynamic threshold from actual readings and the reconstruction and the attention scores.

Figure 36 Indoor temperature sensor readings, the reconstructed time series from the algorithm and the anomaly scores as a demonstration of sequential anomaly in the readings.

Figure 37 provides a comprehensive visualization of the outcomes for each feature, along with the anomalies identified by the algorithm. A notable aggregation of such faults is observable between October 19th and December 23rd. Upon scrutiny of this interval, a significant malfunction within the monitoring system was revealed, impacting all sensors with the exception of those associated with the weather station that records external dry bulb temperature and relative humidity. This malfunction led to shifting of the measurement trend to be linear instead of noisy cyclical.

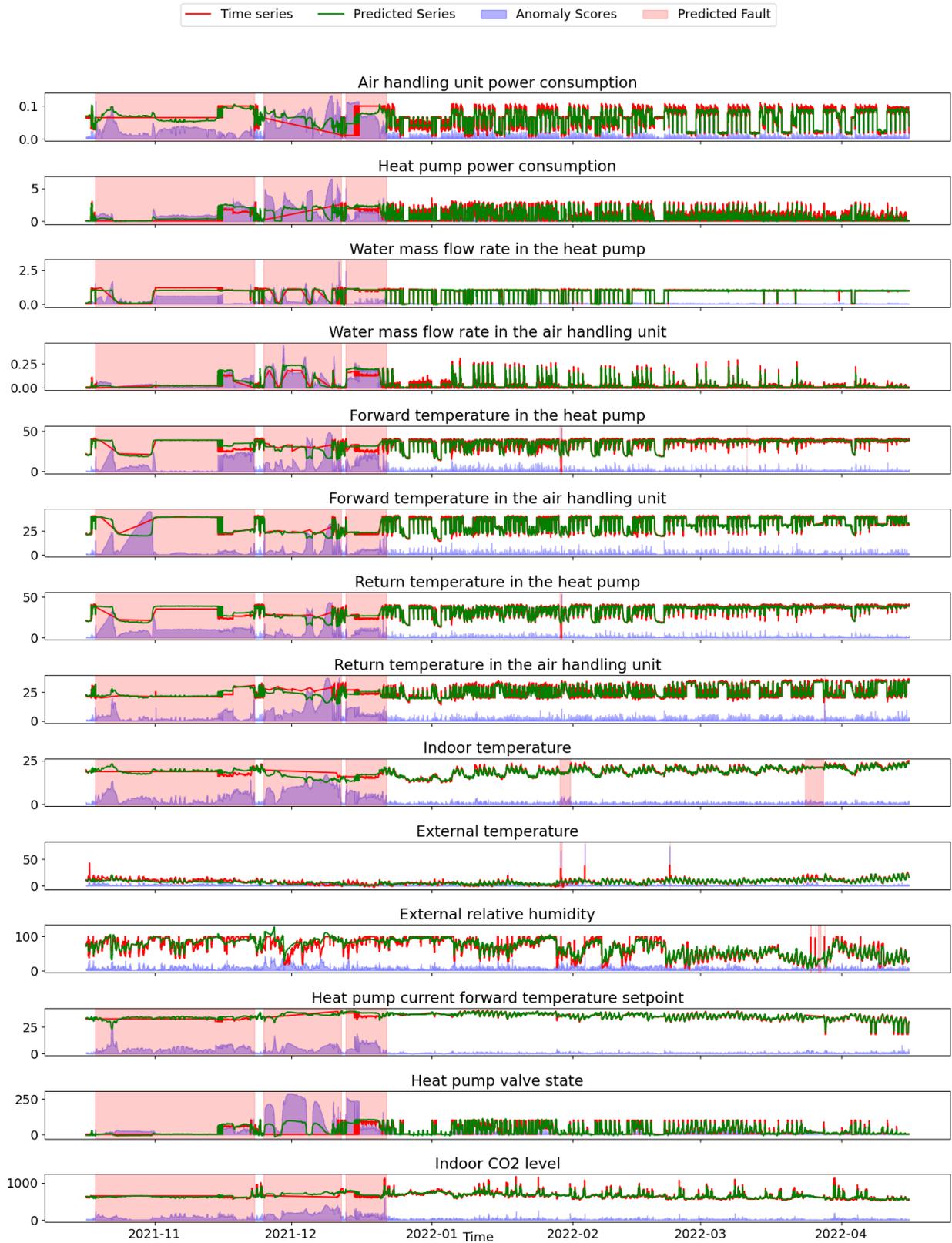


Figure 37 Summary of the outcome for all the features. The highlighted areas are the periods labelled as faults.

One other type of fault that has been identified is related to scheduling patterns shown in Figure 38. The AHU being constantly on during night hours in winter of 2022 triggered a fault. This happened because of the night ventilation that was set during Summer was not turned off during winter.

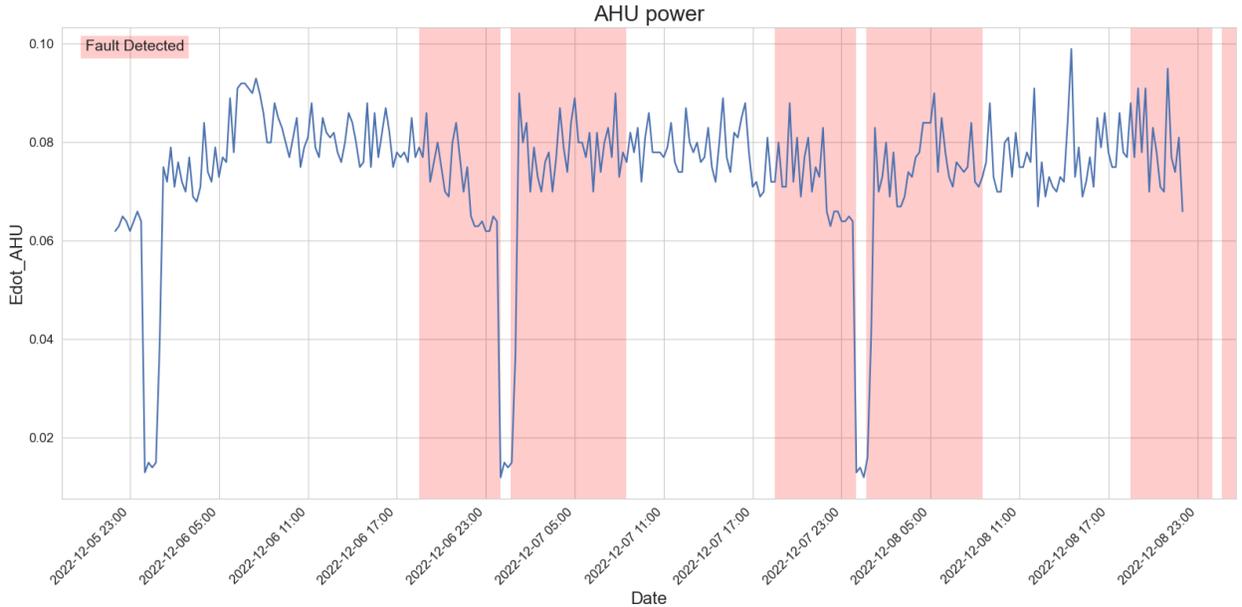


Figure 38: Air handling unit power consumption. Scheduling fault detected is highlighted in red.

Since the model has a window size of 96 which represents one day, the model was able to reconstruct the trends well as shown previously, however, some peaks were not captured in the same precision due to the window size choice. A smaller window choice might solve this issue but will increase the computation cost and compromise capturing the longer trends in different dimensions. A future solution might be to have a dual encoder with different window sizes. Despite the fact that this will lead to increased computational cost, the results should attend to both long and short trends given a correct way of combining the outcome from the dual encoders.

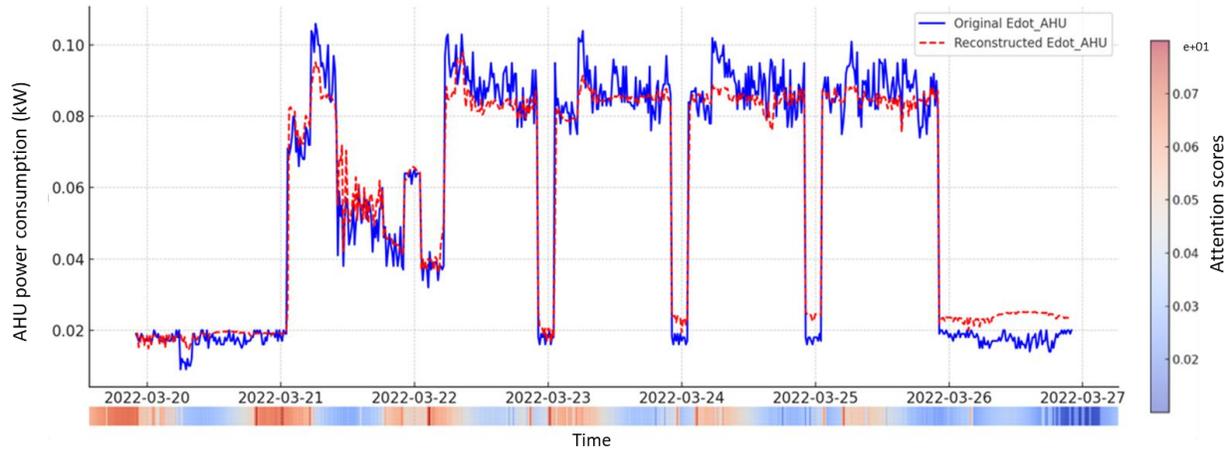


Figure 39 Original time series vs reconstructed time series vs reconstructed for AHU power consumption with average attention scores for every time step.

Figure 39 visualizes the average attention weights of each window averaged over multiple heads. It is apparent that there is a high correlation between the attention weights and peaks and sudden changes in the time series. Analysing the attention weights across different dimensions it was also noticed that the model higher attention weights to different dimensions where the deviations are higher, allowing the model to specifically detect faults in each dimension individually with the contextual trend of the complete sequence as prior.

5 Conclusions

Data-driven fault detection and diagnostics (FDD) in HVAC systems have emerged as a prominent area of research, especially in the context of building management. Despite significant advancements in literature, there remains a noticeable gap between research developments and the availability of fully automated data-driven FDD solutions in the commercial market, particularly within the residential sector. The success of any data-driven application hinges on the quality, availability, and ease of annotation of the data. While the decreasing costs of sensors have facilitated widespread adoption of monitoring systems across various building types, leading to the accumulation of vast datasets, the reliability of this data is often inconsistent. The monitoring systems are frequently plagued by disruptions and faults, whether originating from data pipelines, infrastructure, or the sensors themselves.

Another critical challenge that impedes the widespread adoption of data-driven FDD systems is the scarcity of labelled data for HVAC operations. The annotation process is both labour-intensive and requires domain expertise, unlike fields such as Natural Language Processing (NLP) or Computer Vision (CV), where non-experts can often handle data labelling. This reliance on expert annotation adds a significant barrier to the development and deployment of effective FDD systems.

Much of the existing research has focused on supervised learning algorithms, which require labelled data, and semi-supervised learning, which relies on smaller amounts of labelled data. While these methods are powerful, especially semi-supervised learning where minimal labelled data can yield significant results, their application in real-world systems is limited by their inability to detect or diagnose previously unseen faults. Additionally, these models often function as black boxes, offering little insight into their decision-making processes, which can be problematic for practical applications where interpretability is crucial.

Given these challenges, arguably the immediate goal for most HVAC systems should not be the pursuit of full automation in FDD. While full automation should remain the ultimate objective, the current focus should be on developing highly efficient data collection and labelling pipelines. To this end, It is proposed to decouple the processes of fault detection and fault diagnostics. Fault detection, which typically relies on unsupervised learning, should not depend on labelled data and should be designed to identify unrecognized patterns in the data. Ideally, unsupervised algorithms should be capable of detecting both spatial and temporal patterns across all features. Once these unrecognized patterns are identified, manual labelling should be performed by facility managers or system experts. This approach narrows the scope of the labelling task, making it more manageable and time efficient.

Manual annotation should continue until a sufficient number of faults have been labelled and stored in a dedicated database. Additionally, data from similar systems operating under comparable conditions can be integrated into this fault database, enhancing its robustness. Once an adequate volume of labelled data has been accumulated, fault diagnostics can then be performed using supervised or semi-supervised algorithms. These diagnostic models should be paired with interpretability frameworks and uncertainty quantification mechanisms to ensure transparency in the decision-making process.

In this thesis, The aim was to contribute to both aspects of fault detection and diagnostics. A self-supervised fault detection algorithm that incorporates dynamic detection thresholds was introduced in Chapter 4, and the application of widely used tree-based ensemble algorithms was explored in Chapter 2. Additionally, interpretability frameworks to enhance the transparency of these algorithms was implemented. The feasibility of using multivariate time series classification algorithms for fault diagnostics was assessed in Chapter 3. Through these contributions, I hope to bridge some of the gaps in the current state of FDD research and offer practical insights for future developments in the field.

In the case study, detailed in Chapter 2, The efficacy of machine learning algorithms in detecting and diagnosing faults in a complex, interconnected system was demonstrated. The study utilized a detailed model of a residential apartment, complete with a heat pump, radiant floor heating, and fan coils for cooling. By simulating various fault scenarios, including sensor faults, valve leakages, and pump malfunctions, the research provided a comprehensive dataset for training and evaluating

FDD

algorithms.

The comparative analysis of the most widely used machine learning algorithms, including Random Forest, XGBoost, CatBoost, K-Nearest Neighbours, and Explainable Boosting Machine, offered insights into their relative performance in the context of HVAC fault detection. The results highlighted the superior performance of ensemble methods, particularly Random Forest and XGBoost, in accurately identifying and classifying faults with an initial accuracy of 84% and 83.8% respectively. However, the study also revealed the critical importance of addressing overfitting issues, which were mitigated through careful hyperparameter tuning. In addition, this work displayed the application of interpretability techniques to enhance the transparency and trustworthiness of the FDD models. The use of SHAP (SHapley Additive exPlanations) values provided both global and local interpretations of the model's decision-making process. This approach not only improved the understanding of how the models arrived at their predictions but also offered valuable insights into the relative importance of different features in fault detection. Such interpretability is crucial for gaining the trust of building managers and technicians who may be hesitant to rely on "black box" AI systems for critical infrastructure management. The work also explored briefly the uncertainty in the model's local decisions. The models showed weak performance in some faults and that reflects the high uncertainty in the decision-making process. Examples were shown of predictions with less than 13% confidence.

The research also addressed the challenge of temporal dependencies in HVAC operational data. Chapter 3 presented a comprehensive benchmarking study of multivariate time series classification algorithms for FDD applications. This study filled a crucial gap in the literature by systematically evaluating the performance of various algorithms designed specifically for time series data, including distance-based, interval-based, convolutional-based, and deep learning-based classifiers. The benchmarking study utilized three open-source datasets representing different HVAC subsystems: a boiler plant, a chiller plant, and a single-duct air handling unit. One of the key findings from the benchmarking study was the superior performance of deep learning-based algorithms especially the LSTM-FCN (Long Short-Term Memory Fully Convolutional Network) algorithm across all datasets with F1 scores of 0.98, 0.85 and 0.61 in the three datasets respectively. This result underscores the potential of hybrid deep learning approaches that combine the sequence modelling capabilities of LSTM with the feature extraction power of convolutional neural networks. The study also highlighted the strong performance of the Canonical Interval Forest with

F1 scores of 0.97, 0.82 and 0.62 respectively and ResNet a with F1 scores of .95, 0.82 and 0.62 respectively, suggesting that both traditional machine learning and deep learning approaches have merit in HVAC fault detection. Another aspect of the study was the analysis of computational efficiency alongside accuracy. This dual consideration is crucial for real-world applications, where the trade-off between model performance and computational resources must be carefully balanced. The study revealed significant variations in runtime across different algorithms, with some high-performing models like ResNet requiring substantially more computational resources than faster alternatives like K-Nearest Neighbours with Dynamic Time Warping with a runtime of up to 500 minutes for ResNet and 0.62 for Dynamic Time Warping. Then we compare the performance of the best performant model which is the LSTM-FCN with the ensemble method XGBoost used in chapter 2 to assess whether there is an advantage in performance, the results shows that XGBoost scored F1 score of 0.99, 0.91 and 0.56 in the three datasets respectively which is very similar to the LSTM-FCN with a much lower runtime of 0.14, 0.16 and 0.21 minutes respectively.

Building upon the insights gained from the supervised learning approaches, Chapter 4 introduced a novel self-supervised transformer-based architecture for fault detection in HVAC systems. This approach addresses one of the fundamental challenges in the field: the scarcity of labelled fault data. By leveraging the power of self-supervised learning, the proposed method can learn meaningful representations from unlabelled time series data, making it particularly valuable for real-world applications where annotated fault data is limited or non-existent. The self-supervised transformer architecture incorporates several key innovations. First, it adapts the transformer model, originally designed for natural language processing tasks, to the specific challenges of multivariate time series data in HVAC systems. The use of learnable positional encodings and the integration of the Time2Vec method for encoding timestamps allowed the model to effectively capture both short-term and long-term temporal dependencies in the data. A crucial aspect of the proposed architecture is its encoder-only design, which streamlines the model and improves computational efficiency compared to full encoder-decoder architectures. This design choice is particularly well-suited to the task of anomaly detection, where the goal is to identify deviations from normal patterns rather than generate new sequences. The self-supervised pre-training approach, based on an autoregressive masking task, enables the model to learn robust representations of normal system behaviour without requiring labelled fault data. This pre-training step is a key innovation that allows the model to generalize well to unseen data and potentially

detect novel fault types that were not present in the training data. The integration of a dynamic thresholding technique based on extreme value theory further enhances the model's ability to adapt to changing operational conditions and detect anomalies with high precision. The use of the Peak Over Threshold method, coupled with the Generalized Pareto Distribution, provides a statistically rigorous approach to setting anomaly detection thresholds that can automatically adjust to the characteristics of the data. To validate the effectiveness of the proposed approach, the model was applied to real-world data from a Nearly Zero Energy Building equipped with a complex HVAC system. The results demonstrated the model's ability to accurately reconstruct normal system behaviour and identify both point anomalies and more subtle sequential anomalies. Most of the faults detected were related to the monitoring system. Those faults detected were both point anomalies and sequential ones. Also scheduling faults were detected. The model's attention mechanism provided additional insights into which aspects of the multivariate time series were most relevant for detecting anomalies at different time points. The case study also highlighted some limitations and areas for future improvement. For instance, the fixed window size used in the current implementation may not be optimal for capturing both short-term and long-term patterns simultaneously. Future work could explore multi-scale approaches or adaptive window sizes to address this limitation.

As for future area of research uncertainty quantification represents a crucial next step in enhancing the reliability and trustworthiness of FDD systems. While the current work has focused on improving the accuracy and interpretability of fault detection models, quantifying the uncertainty associated with these predictions is essential for practical implementation. Future research should explore the application of frameworks such as conformal prediction to FDD models. Conformal prediction offers a mathematically rigorous approach to constructing prediction intervals with guaranteed coverage probabilities, regardless of the underlying distribution of the data. This could provide building managers with valuable information about the confidence level of fault predictions, enabling more informed decision-making. For instance, in cases where the model predicts a fault but with high uncertainty, managers might opt for additional manual inspections rather than immediately initiating costly maintenance procedures. Moreover, uncertainty quantification could help in identifying areas where the model's knowledge is limited, potentially indicating the need for additional training data or highlighting novel fault types that the model is not yet equipped to handle confidently. Integrating these uncertainty estimates with the

interpretability techniques explored in this thesis could offer a more comprehensive understanding of the model's decision-making process and limitations. Another critical area for future research lies in addressing three fundamental challenges that impact the practical adoption of FDD systems: transferability, portability, and deployment. The transferability of FDD models across different buildings and HVAC systems requires innovative approaches to develop more generalizable solutions. Future research should explore transfer learning techniques that can leverage knowledge gained from well-monitored buildings to enhance FDD performance in buildings with limited historical data. Additionally, the development of building-agnostic feature extraction methods that capture universal fault patterns rather than building-specific characteristics could significantly improve model transferability. The portability challenge presents another promising research direction, specifically in creating standardized frameworks that can accommodate diverse HVAC configurations and sensor arrangements. Research efforts should focus on developing adaptive architectures that can automatically adjust to different system configurations and sensor types without requiring extensive reconfiguration or retraining. Regarding deployment, future work should investigate efficient methods for integrating FDD systems with existing building management infrastructure. This includes developing standardized protocols for real-time data processing, exploring edge computing solutions to reduce computational overhead, and creating scalable architectures that can efficiently handle multiple buildings simultaneously. Research in these areas would bridge the current gap between theoretical FDD models and their practical implementation, ultimately facilitating wider adoption of these systems in real-world applications.

Bibliography

- [1] I. Hamilton and O. Rapf, “E2020 global status report for buildings and construction: towards a zero-emissions, efficient and resilient buildings and construction sector,” *Global Alliance for Buildings and Construction*, pp. 1–7, 2020.
- [2] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” *Energy Build*, vol. 40, no. 3, pp. 394–398, 2008, doi: 10.1016/j.enbuild.2007.03.007.
- [3] M. R. Brambley and S. Katipamula, “Commercial Building Retuning A Low-Cost Way to Improve Energy Performance,” *ASHRAE J*, vol. 51, no. October, 2009.
- [4] K. W. Roth, D. Westphalen, P. Llana, and M. Feng, “The Energy Impact of Faults in US Commercial Buildings,” *International Refrigeration and Air Conditioning Conference*, pp. 600–609, 2004.
- [5] G. Lin, H. Kramer, and J. Granderson, “Building fault detection and diagnostics: Achieved savings, and methods to evaluate algorithm performance,” *Build Environ*, vol. 168, no. October 2019, p. 106505, 2020, doi: 10.1016/j.buildenv.2019.106505.
- [6] W. Kim and S. Katipamula, “A review of fault detection and diagnostics methods for building systems,” *Sci Technol Built Environ*, vol. 24, no. 1, pp. 3–21, 2018, doi: 10.1080/23744731.2017.1318008.
- [7] S. Katipamula and M. R. Brambley, “Review article: Methods for fault detection, diagnostics, and prognostics for building systems—a review, part II,” *HVAC and R Research*, vol. 11, no. 2, pp. 169–187, 2005, doi: 10.1080/10789669.2005.10391133.
- [8] S. Katipamula, M. R. Brambley, and M. R. Brambley, “Methods for Fault Detection , Diagnostics , and Prognostics for Building Systems — A Review , Part I,” *HVAC and R Research*, vol. 11, no. 1, pp. 3–25, 2005.
- [9] S. Frank, X. Jin, D. Studer, and A. Farthing, “Assessing barriers and research challenges for automated fault detection and diagnosis technology for small commercial buildings in the

- United States,” *Renewable and Sustainable Energy Reviews*, vol. 98, no. January, pp. 489–499, 2018, doi: 10.1016/j.rser.2018.08.046.
- [10] N. Omri, Z. Al Masry, N. Mairrot, S. Giampiccolo, and N. Zerhouni, “Towards an adapted PHM approach: Data quality requirements methodology for fault detection applications,” *Comput Ind*, vol. 127, p. 103414, 2021, doi: 10.1016/j.compind.2021.103414.
- [11] Y. Li and Z. O’Neill, “A critical review of fault modeling of HVAC systems in buildings,” *Build Simul*, vol. 11, no. 5, pp. 953–975, 2018, doi: 10.1007/s12273-018-0458-4.
- [12] Z. Shi and W. O’Brien, “Development and implementation of automated fault detection and diagnostics for building systems: A review,” *Autom Constr*, vol. 104, no. March, pp. 215–229, 2019, doi: 10.1016/j.autcon.2019.04.002.
- [13] S. Pourarian, J. Wen, D. Veronica, A. Pertzborn, X. Zhou, and R. Liu, “A tool for evaluating fault detection and diagnostic methods for fan coil units,” *Energy Build*, vol. 136, pp. 151–160, 2017, doi: 10.1016/j.enbuild.2016.12.018.
- [14] R. Chiosa, M. S. Piscitelli, and A. Capozzoli, “A data analytics-based energy information system (EIS) tool to perform meter-level anomaly detection and diagnosis in buildings,” *Energies (Basel)*, vol. 14, no. 1, 2021, doi: 10.3390/en14010237.
- [15] M. S. Piscitelli, S. Brandi, A. Capozzoli, and F. Xiao, “A data analytics-based tool for the detection and diagnosis of anomalous daily energy patterns in buildings,” *Build Simul*, vol. 14, no. 1, pp. 131–147, 2021, doi: 10.1007/s12273-020-0650-1.
- [16] I. Matetić, I. Štajduhar, I. Wolf, and S. Ljubic, “A Review of Data-Driven Approaches and Techniques for Fault Detection and Diagnosis in HVAC Systems,” *Sensors*, vol. 23, no. 1, p. 1, Jan. 2023, doi: 10.3390/S23010001/S1.
- [17] F. Calabrese, A. Regattieri, M. Bortolini, and F. G. Galizia, “Data-Driven Fault Detection and Diagnosis: Challenges and Opportunities in Real-World Scenarios,” *Applied Sciences* 2022, Vol. 12, Page 9212, vol. 12, no. 18, p. 9212, Sep. 2022, doi: 10.3390/APP12189212.
- [18] K. Chen, S. Chen, X. Zhu, X. Jin, and Z. Du, “Interpretable mechanism mining enhanced deep learning for fault diagnosis of heating, ventilation and air conditioning systems,” *Build Environ*, vol. 237, p. 110328, Jun. 2023, doi: 10.1016/J.BUILDENV.2023.110328.

- [19] Z. Chen *et al.*, “A review of data-driven fault detection and diagnostics for building HVAC systems,” *Appl Energy*, vol. 339, p. 121030, Jun. 2023, doi: 10.1016/J.APENERGY.2023.121030.
- [20] T. Li, Y. Zhao, C. Zhang, J. Luo, and X. Zhang, “A knowledge-guided and data-driven method for building HVAC systems fault diagnosis,” *Build Environ*, vol. 198, p. 107850, Jul. 2021, doi: 10.1016/J.BUILDENV.2021.107850.
- [21] R. Zhang and T. Hong, “Modeling of HVAC operational faults in building performance simulation,” *Appl Energy*, vol. 202, pp. 178–188, Sep. 2017, doi: 10.1016/J.APENERGY.2017.05.153.
- [22] Y. Zhao, S. Wang, F. Xiao, and Z. Ma, “A simplified physical model-based fault detection and diagnosis strategy and its customized tool for centrifugal chillers,” *HVAC&R Res*, vol. 19, no. 3, pp. 283–294, Apr. 2013, doi: 10.1080/10789669.2013.765299.
- [23] T. Li, M. Deng, Y. Zhao, X. Zhang, and C. Zhang, “An air handling unit fault isolation method by producing additional diagnostic information proactively,” *Sustainable Energy Technologies and Assessments*, vol. 43, p. 100953, Feb. 2021, doi: 10.1016/J.SETA.2020.100953.
- [24] Y. Zhao *et al.*, “A proactive fault detection and diagnosis method for variable-air-volume terminals in building air conditioning systems,” *Energy Build*, vol. 183, pp. 527–537, Jan. 2019, doi: 10.1016/J.ENBUILD.2018.11.021.
- [25] J. Chen, L. Zhang, Y. Li, Y. Shi, X. Gao, and Y. Hu, “A review of computing-based automated fault detection and diagnosis of heating, ventilation and air conditioning systems,” *Renewable and Sustainable Energy Reviews*, vol. 161, p. 112395, Jun. 2022, doi: 10.1016/J.RSER.2022.112395.
- [26] 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, Jul. 2019, doi: 10.1016/J.RSER.2019.04.021.

- [27] M. S. Mirnaghi and F. Haghghat, "Fault detection and diagnosis of large-scale HVAC systems in buildings using data-driven methods: A comprehensive review," *Energy Build*, vol. 229, Dec. 2020, doi: 10.1016/j.enbuild.2020.110492.
- [28] "Towards a Scalable Model for Smart Buildings | Building Technology and Urban Systems." Accessed: Aug. 10, 2024. [Online]. Available: <https://buildings.lbl.gov/publications/towards-scalable-model-smart>
- [29] Z. Jin, W. Yezheng, and Y. Gang, "A stochastic method to generate bin weather data in Nanjing, China," *Energy Convers Manag*, vol. 47, no. 13–14, pp. 1843–1850, Aug. 2006, doi: 10.1016/J.ENCONMAN.2005.10.006.
- [30] A. A. Kasam, B. D. Lee, and C. J. J. Paredis, "Statistical methods for interpolating missing meteorological data for use in building simulation," *Build Simul*, vol. 7, no. 5, pp. 455–465, Mar. 2014, doi: 10.1007/S12273-014-0174-7/METRICS.
- [31] A. Rahman, V. Srikumar, and A. D. Smith, "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks," *Appl Energy*, vol. 212, pp. 372–385, Feb. 2018, doi: 10.1016/j.apenergy.2017.12.051.
- [32] J. Yang, K. K. Tan, M. Santamouris, and S. E. Lee, "Building Energy Consumption Raw Data Forecasting Using Data Cleaning and Deep Recurrent Neural Networks," *Buildings 2019, Vol. 9, Page 204*, vol. 9, no. 9, p. 204, Sep. 2019, doi: 10.3390/BUILDINGS9090204.
- [33] D. Inman, R. Elmore, and B. Bush, "A case study to examine the imputation of missing data to improve clustering analysis of building electrical demand," *Building Services Engineering Research and Technology*, vol. 36, no. 5, pp. 628–637, Sep. 2015, doi: 10.1177/0143624415573215.
- [34] L. Zhang, "A pattern-recognition-based ensemble data imputation framework for sensors from building energy systems," *Sensors (Switzerland)*, vol. 20, no. 20, pp. 1–16, Oct. 2020, doi: 10.3390/S20205947.
- [35] D. Li, Y. Zhou, G. Hu, and C. J. Spanos, "Handling Incomplete Sensor Measurements in Fault Detection and Diagnosis for Building HVAC Systems," *IEEE Transactions on*

- Automation Science and Engineering*, vol. 17, no. 2, pp. 833–846, Apr. 2020, doi: 10.1109/TASE.2019.2948101.
- [36] Z. Wang, L. Wang, Y. Tan, and J. Yuan, “Fault detection based on Bayesian network and missing data imputation for building energy systems,” *Appl Therm Eng*, vol. 182, Jan. 2021, doi: 10.1016/j.applthermaleng.2020.116051.
- [37] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,” *Front Energy Res*, vol. 9, p. 652801, Mar. 2021, doi: 10.3389/FENRG.2021.652801/BIBTEX.
- [38] L. Zhang, S. Frank, J. Kim, X. Jin, and M. Leach, “A systematic feature extraction and selection framework for data-driven whole-building automated fault detection and diagnostics in commercial buildings,” *Build Environ*, vol. 186, Dec. 2020, doi: 10.1016/j.buildenv.2020.107338.
- [39] L. Zhang and J. Wen, “A systematic feature selection procedure for short-term data-driven building energy forecasting model development,” *Energy Build*, vol. 183, pp. 428–442, Jan. 2019, doi: 10.1016/j.enbuild.2018.11.010.
- [40] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.
- [41] D. Li, Y. Zhou, G. Hu, and C. J. Spanos, “Optimal Sensor Configuration and Feature Selection for AHU Fault Detection and Diagnosis,” *IEEE Trans Industr Inform*, vol. 13, no. 3, pp. 1369–1380, Jun. 2017, doi: 10.1109/TII.2016.2644669.
- [42] S. Li and J. Wen, “Development and validation of a dynamic air handling unit model, Part I,” *ASHRAE Trans*, vol. 116, no. 1, pp. 45–57, Jan. 2010, Accessed: Aug. 10, 2024. [Online]. Available: <https://go.gale.com/ps/i.do?p=AONE&sw=w&issn=00012505&v=2.1&it=r&id=GALE%7CA227975372&sid=googleScholar&linkaccess=fulltext>

- [43] S. Li, J. Wen, X. Zhou, C. J. Klaassen, and ASHRAE, “Development and Validation of a Dynamic Air Handling Unit Model, Part 2,” *ASHRAE TRANSACTIONS 2010, VOL 116, PT 1*, vol. 116, no. 1, 2010, Accessed: Aug. 10, 2024. [Online]. Available: <https://researchdiscovery.drexel.edu/esploro/outputs/991019170343204721>
- [44] S. M. Namburu, M. S. Azam, 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, Jul. 2007, doi: 10.1109/TASE.2006.888053.
- [45] R. Yan, Z. Ma, Y. Zhao, and G. Kokogiannakis, “A decision tree based data-driven diagnostic strategy for air handling units,” *Energy Build*, vol. 133, pp. 37–45, Dec. 2016, doi: 10.1016/j.enbuild.2016.09.039.
- [46] A. Barredo Arrieta *et al.*, “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/J.INFFUS.2019.12.012.
- [47] A. Rosato, F. Guarino, S. Sibilio, E. Entchev, M. Masullo, and L. Maffei, “Healthy and Faulty Experimental Performance of a Typical HVAC System under Italian Climatic Conditions: Artificial Neural Network-Based Model and Fault Impact Assessment,” *Energies 2021, Vol. 14, Page 5362*, vol. 14, no. 17, p. 5362, Aug. 2021, doi: 10.3390/EN14175362.
- [48] S. Miyata, Y. Akashi, J. Lim, Y. Kuwahara, and K. Tanaka, “Model-based Fault Detection and Diagnosis for HVAC Systems Using Convolutional Neural Network,” *Building Simulation Conference Proceedings*, vol. 16, pp. 853–860, Sep. 2019, doi: 10.26868/25222708.2019.210311.
- [49] J. Aguilar, A. Garces-Jimenez, J. M. Gomez-Pulido, M. D. R. Moreno, J. A. G. De Mesa, and N. Gallego-Salvador, “Autonomic Management of a Building’s Multi-HVAC System Start-Up,” *IEEE Access*, vol. 9, pp. 70502–70515, 2021, doi: 10.1109/ACCESS.2021.3078550.

- [50] X. Zhu, K. Chen, B. Anduv, X. Jin, and Z. Du, “Transfer learning based methodology for migration and application of fault detection and diagnosis between building chillers for improving energy efficiency,” *Build Environ*, vol. 200, p. 107957, Aug. 2021, doi: 10.1016/J.BUILDENV.2021.107957.
- [51] M. Elnour and N. Meskin, “Novel Actuator Fault Diagnosis Framework for Multizone HVAC Systems Using 2-D Convolutional Neural Networks,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1985–1996, Jul. 2022, doi: 10.1109/TASE.2021.3067866.
- [52] F. Cheng, W. Cai, X. Zhang, H. Liao, and C. Cui, “Fault detection and diagnosis for Air Handling Unit based on multiscale convolutional neural networks,” *Energy Build*, vol. 236, p. 110795, Apr. 2021, doi: 10.1016/J.ENBUILD.2021.110795.
- [53] G. Li *et al.*, “An explainable one-dimensional convolutional neural networks based fault diagnosis method for building heating, ventilation and air conditioning systems,” *Build Environ*, vol. 203, p. 108057, Oct. 2021, doi: 10.1016/J.BUILDENV.2021.108057.
- [54] S. Taheri, A. Ahmadi, B. Mohammadi-Ivatloo, and S. Asadi, “Fault detection diagnostic for HVAC systems via deep learning algorithms,” *Energy Build*, vol. 250, p. 111275, Nov. 2021, doi: 10.1016/J.ENBUILD.2021.111275.
- [55] M. S. Piscitelli, S. Brandi, A. Capozzoli, and F. Xiao, “A data analytics-based tool for the detection and diagnosis of anomalous daily energy patterns in buildings,” *Build Simul*, vol. 14, no. 1, pp. 131–147, Feb. 2021, doi: 10.1007/S12273-020-0650-1.
- [56] S. Gharsellaoui, M. Mansouri, S. S. Refaat, H. Abu-Rub, and H. Messaoud, “Multivariate Features Extraction and Effective Decision Making Using Machine Learning Approaches,” *Energies 2020, Vol. 13, Page 609*, vol. 13, no. 3, p. 609, Jan. 2020, doi: 10.3390/EN13030609.
- [57] W. Kim and J. E. Braun, “Development, implementation, and evaluation of a fault detection and diagnostics system based on integrated virtual sensors and fault impact models,” *Energy Build*, vol. 228, p. 110368, Dec. 2020, doi: 10.1016/J.ENBUILD.2020.110368.

- [58] C. P. Dowling and B. Zhang, “Transfer Learning for HVAC System Fault Detection,” *Proceedings of the American Control Conference*, vol. 2020-July, pp. 3879–3885, Jul. 2020, doi: 10.23919/ACC45564.2020.9147772.
- [59] A. Taal, L. Itard, and W. Zeiler, “A diagnostic Bayesian network method to diagnose building energy performance,” *Building Simulation Conference Proceedings*, vol. 2, pp. 893–899, Sep. 2019, doi: 10.26868/25222708.2019.210945.
- [60] A. Taal and L. Itard, “Fault detection and diagnosis for indoor air quality in DCV systems: Application of 4S3F method and effects of DBN probabilities,” *Build Environ*, vol. 174, p. 106632, May 2020, doi: 10.1016/J.BUILDENV.2019.106632.
- [61] C. Yang, B. Gunay, Z. Shi, and W. Shen, “Machine Learning-Based Prognostics for Central Heating and Cooling Plant Equipment Health Monitoring,” *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 346–355, Jan. 2021, doi: 10.1109/TASE.2020.2998586.
- [62] S. Zhang, X. Zhu, B. Anduv, X. Jin, and Z. Du, “Fault detection and diagnosis for the screw chillers using multi-region XGBoost model,” *Sci Technol Built Environ*, vol. 27, no. 5, pp. 608–623, 2021, doi: 10.1080/23744731.2021.1877966.
- [63] D. Chakraborty and H. Elzarka, “Early detection of faults in HVAC systems using an XGBoost model with a dynamic threshold,” *Energy Build*, vol. 185, pp. 326–344, Feb. 2019, doi: 10.1016/J.ENBUILD.2018.12.032.
- [64] W. S. Yun, W. H. Hong, and H. Seo, “A data-driven fault detection and diagnosis scheme for air handling units in building HVAC systems considering undefined states,” *Journal of Building Engineering*, vol. 35, p. 102111, Mar. 2021, doi: 10.1016/J.JOBE.2020.102111.
- [65] N. Tabassam, S. Amin, and R. Obermaisser, “Fault detection and diagnosis in hvac systems using diagnostic multi-query graphs,” *Proceedings - 2019 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking, ISPA/BDCloud/SustainCom/SocialCom 2019*, pp. 628–633, Dec. 2019, doi: 10.1109/ISPA-BDCLOUD-SUSTAINCOM-SOCIALCOM48970.2019.00095.

- [66] H. Han, Z. Zhang, X. Cui, and Q. Meng, “Ensemble learning with member optimization for fault diagnosis of a building energy system,” *Energy Build*, vol. 226, p. 110351, Nov. 2020, doi: 10.1016/J.ENBUILD.2020.110351.
- [67] A. Gálvez, A. Diez-Olivan, D. Seneviratne, and D. Galar, “Fault Detection and RUL Estimation for Railway HVAC Systems Using a Hybrid Model-Based Approach,” *Sustainability 2021, Vol. 13, Page 6828*, vol. 13, no. 12, p. 6828, Jun. 2021, doi: 10.3390/SU13126828.
- [68] I. H. Witten, E. Frank, and J. Geller, “Data mining,” *ACM SIGMOD Record*, vol. 31, no. 1, pp. 76–77, Mar. 2002, doi: 10.1145/507338.507355.
- [69] M. S. Piscitelli, D. M. Mazzarelli, and A. Capozzoli, “Enhancing operational performance of AHUs through an advanced fault detection and diagnosis process based on temporal association and decision rules,” *Energy Build*, vol. 226, Nov. 2020, doi: 10.1016/j.enbuild.2020.110369.
- [70] L. Chen, H. B. Gunay, Z. Shi, W. Shen, and X. Li, “A Metadata inference method for building automation systems with limited semantic information,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 2107–2119, Oct. 2020, doi: 10.1109/TASE.2020.2990566.
- [71] Y. Xu *et al.*, “An anomaly detection and dynamic energy performance evaluation method for HVAC systems based on data mining,” *Sustainable Energy Technologies and Assessments*, vol. 44, p. 101092, Apr. 2021, doi: 10.1016/J.SETA.2021.101092.
- [72] A. Ranade, G. Provan, A. El-Din Mady, and D. O’Sullivan, “A computationally efficient method for fault diagnosis of fan-coil unit terminals in building Heating Ventilation and Air Conditioning systems,” *Journal of Building Engineering*, vol. 27, p. 100955, Jan. 2020, doi: 10.1016/J.JOBE.2019.100955.
- [73] M. Parzinger, U. Wellisch, L. Hanfstaengl, F. Sigg, M. Wirnsberger, and U. Spindler, “Identifying faults in the building system based on model prediction and residuum analysis,” *E3S Web of Conferences*, vol. 172, p. 22001, Jun. 2020, doi: 10.1051/E3SCONF/202017222001.

- [74] M. Parzinger, L. Hanfstaengl, F. Sigg, U. Spindler, U. Wellisch, and M. Wirnsberger, “Residual Analysis of Predictive Modelling Data for Automated Fault Detection in Building’s Heating, Ventilation and Air Conditioning Systems,” *Sustainability* 2020, Vol. 12, Page 6758, vol. 12, no. 17, p. 6758, Aug. 2020, doi: 10.3390/SU12176758.
- [75] B. Wu, W. Cai, H. Chen, and X. Zhang, “A hybrid data-driven simultaneous fault diagnosis model for air handling units,” *Energy Build*, vol. 245, p. 111069, Aug. 2021, doi: 10.1016/J.ENBUILD.2021.111069.
- [76] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785/SUPPL_FILE/KDD2016_CHEN_BOOSTING_SYSTEM_01-ACM.MP4.
- [77] M. Dey, S. P. Rana, and S. Dudley, “A case study based approach for remote fault detection using multi-level machine learning in a smart building,” *Smart Cities*, vol. 3, no. 2, pp. 401–419, Jun. 2020, doi: 10.3390/SMARTCITIES3020021.
- [78] H. B. Gunay and Z. Shi, “Cluster analysis-based anomaly detection in building automation systems,” *Energy Build*, vol. 228, p. 110445, Dec. 2020, doi: 10.1016/J.ENBUILD.2020.110445.
- [79] A. A. Markus, B. W. Hobson, H. B. Gunay, and S. Bucking, “A framework for a multi-source, data-driven building energy management toolkit,” *Energy Build*, vol. 250, p. 111255, Nov. 2021, doi: 10.1016/J.ENBUILD.2021.111255.
- [80] C. Zhang, X. Xue, Y. Zhao, X. Zhang, and T. Li, “An improved association rule mining-based method for revealing operational problems of building heating, ventilation and air conditioning (HVAC) systems,” *Appl Energy*, vol. 253, p. 113492, Nov. 2019, doi: 10.1016/J.APENERGY.2019.113492.
- [81] C. Zhang, Y. Zhao, J. Lu, T. Li, and X. Zhang, “Analytic hierarchy process-based fuzzy post mining method for operation anomaly detection of building energy systems,” *Energy Build*, vol. 252, p. 111426, Dec. 2021, doi: 10.1016/J.ENBUILD.2021.111426.

- [82] H. Zhang, C. Li, D. Li, Y. Zhang, and W. Peng, “Fault detection and diagnosis of the air handling unit via an enhanced kernel slow feature analysis approach considering the time-wise and batch-wise dynamics,” *Energy Build*, vol. 253, p. 111467, Dec. 2021, doi: 10.1016/J.ENBUILD.2021.111467.
- [83] K. Yan, J. Huang, W. Shen, and Z. Ji, “Unsupervised learning for fault detection and diagnosis of air handling units,” *Energy Build*, vol. 210, p. 109689, Mar. 2020, doi: 10.1016/J.ENBUILD.2019.109689.
- [84] Y. Bouabdallaoui, Z. Lafhaj, P. Yim, L. Ducoulombier, and B. Bennadji, “Predictive Maintenance in Building Facilities: A Machine Learning-Based Approach,” *Sensors 2021, Vol. 21, Page 1044*, vol. 21, no. 4, p. 1044, Feb. 2021, doi: 10.3390/S21041044.
- [85] Y. Chen, G. Lin, E. Crowe, and J. Granderson, “Development of a Unified Taxonomy for HVAC System Faults,” *Energies 2021, Vol. 14, Page 5581*, vol. 14, no. 17, p. 5581, Sep. 2021, doi: 10.3390/EN14175581.
- [86] C. P. Dowling and B. Zhang, “Transfer Learning for HVAC System Fault Detection,” *Proceedings of the American Control Conference*, vol. 2020-July, pp. 3879–3885, Jul. 2020, doi: 10.23919/ACC45564.2020.9147772.
- [87] M. Lampis and J. D. Andrews, “Bayesian belief networks for system fault diagnostics,” *Qual Reliab Eng Int*, vol. 25, no. 4, pp. 409–426, Jun. 2009, doi: 10.1002/QRE.978.
- [88] H. Olsson *et al.*, “Estimating diagnostic uncertainty in artificial intelligence assisted pathology using conformal prediction,” *Nature Communications 2022 13:1*, vol. 13, no. 1, pp. 1–10, Dec. 2022, doi: 10.1038/s41467-022-34945-8.
- [89] J. B. Butzbaugh, A. S. D. Tidwell, and C. A. Antonopoulos, “Automatic Fault Detection & Diagnostics: Residential Market Analysis,” Sep. 2020, doi: 10.2172/1670423.
- [90] K. A. Ejenakevwe and L. Song, “Review of Fault Detection and Diagnosis Studies on Residential HVAC Systems,” *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, vol. 8B-2021, Jan. 2022, doi: 10.1115/IMECE2021-72745.
- [91] “Modelica Buildings library.”

- [92] “GitHub - ibpsa/modelica-ibpsa: Modelica library for building and district energy systems developed within IBPSA Project 1.”
- [93] J. Bi, H. Wang, M. Hua, and K. Yan, “An interpretable feature selection method integrating ensemble models for chiller fault diagnosis,” *Journal of Building Engineering*, vol. 87, p. 109029, Jun. 2024, doi: 10.1016/J.JOBE.2024.109029.
- [94] A. E. Maxwell, M. Sharma, and K. A. Donaldson, “Explainable Boosting Machines for Slope Failure Spatial Predictive Modeling,” *Remote Sensing 2021, Vol. 13, Page 4991*, vol. 13, no. 24, p. 4991, Dec. 2021, doi: 10.3390/RS13244991.
- [95] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, “Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization,” *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, Mar. 2019, doi: 10.11989/JEST.1674-862X.80904120.
- [96] “parrr/dtreviz: A python library for decision tree visualization and model interpretation.” Accessed: Aug. 15, 2024. [Online]. Available: <https://github.com/parrr/dtreviz>
- [97] S. M. Lundberg and S. I. Lee, “A Unified Approach to Interpreting Model Predictions,” *Adv Neural Inf Process Syst*, vol. 2017-December, pp. 4766–4775, May 2017, Accessed: Aug. 16, 2024. [Online]. Available: <https://arxiv.org/abs/1705.07874v2>
- [98] Z. Wang, Y. Dong, W. Liu, and Z. Ma, “A Novel Fault Diagnosis Approach for Chillers Based on 1-D Convolutional Neural Network and Gated Recurrent Unit,” *Sensors 2020, Vol. 20, Page 2458*, vol. 20, no. 9, p. 2458, Apr. 2020, doi: 10.3390/S20092458.
- [99] Z. Jiang, M. J. Risbeck, S. C. Kulandai Samy, C. Zhang, S. Cyrus, and Y. M. Lee, “A timeseries supervised learning framework for fault prediction in chiller systems,” *Energy Build*, vol. 285, p. 112876, Apr. 2023, doi: 10.1016/J.ENBUILD.2023.112876.
- [100] A. P. Ruiz, M. Flynn, and A. Bagnall, “Benchmarking Multivariate Time Series Classification Algorithms,” *Data Min Knowl Discov*, vol. 35, no. 2, pp. 401–449, Jul. 2020, doi: 10.1007/s10618-020-00727-3.
- [101] M. Middlehurst, J. Large, and A. Bagnall, “The Canonical Interval Forest (CIF) Classifier for Time Series Classification,” *Proceedings - 2020 IEEE International Conference on Big*

- Data, Big Data* 2020, pp. 188–195, Dec. 2020, doi: 10.1109/BIGDATA50022.2020.9378424.
- [102] H. Deng, G. Runger, E. Tuv, and M. Vladimir, “A Time Series Forest for Classification and Feature Extraction,” *Inf Sci (N Y)*, vol. 239, pp. 142–153, Feb. 2013, doi: 10.1016/j.ins.2013.02.030.
- [103] C. H. Lubba *et al.*, “catch22: CAnonical Time-series CHaracteristics,” *Data Min Knowl Discov*, vol. 33, no. 6, pp. 1821–1852, Jan. 2019, doi: 10.1007/s10618-019-00647-x.
- [104] A. Dempster, F. Petitjean, and G. I. Webb, “ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Min Knowl Discov*, vol. 34, no. 5, pp. 1454–1495, Oct. 2019, doi: 10.1007/s10618-020-00701-z.
- [105] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, “HIVE-COTE 2.0: a new meta ensemble for time series classification,” *Mach Learn*, vol. 110, no. 11–12, pp. 3211–3243, Apr. 2021, doi: 10.1007/s10994-021-06057-9.
- [106] C. Peng and Q. Cheng, “Discriminative Ridge Machine: A Classifier for High-Dimensional Data or Imbalanced Data,” *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 6, pp. 2595–2609, Apr. 2019, doi: 10.1109/TNNLS.2020.3006877.
- [107] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, “Deep learning for time series classification: a review,” *Data Min Knowl Discov*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/S10618-019-00619-1/FIGURES/16.
- [108] X. Huang, C. Zhu, and W. Chen, “RestNet: Boosting Cross-Domain Few-Shot Segmentation with Residual Transformation Network,” Aug. 2023, Accessed: Apr. 20, 2024. [Online]. Available: <https://arxiv.org/abs/2308.13469v2>
- [109] F. Karim, S. Majumdar, H. Darabi, and S. Harford, “Multivariate LSTM-FCNs for Time Series Classification,” *Neural Networks*, vol. 116, pp. 237–245, Jan. 2018, doi: 10.1016/j.neunet.2019.04.014.
- [110] M. Abouelnaga, J. Vitay, and A. Farahani, “Multivariate Time Series Classification: A Deep Learning Approach,” Jul. 2023, Accessed: Apr. 20, 2024. [Online]. Available: <https://arxiv.org/abs/2307.02253v1>

- [111] H. I. Fawaz *et al.*, “InceptionTime: Finding AlexNet for Time Series Classification,” *Data Min Knowl Discov*, vol. 34, no. 6, pp. 1936–1962, Sep. 2019, doi: 10.1007/s10618-020-00710-y.
- [112] N. Tabassum, S. Menon, and A. Jastrzębska, “Time-series classification with SAFE: Simple and fast segmented word embedding-based neural time series classifier,” *Inf Process Manag*, vol. 59, no. 5, p. 103044, Sep. 2022, doi: 10.1016/J.IPM.2022.103044.
- [113] M. Middlehurst, P. Schäfer, and A. Bagnall, “Bake off redux: a review and experimental evaluation of recent time series classification algorithms,” Apr. 2023, Accessed: Apr. 21, 2024. [Online]. Available: <https://arxiv.org/abs/2304.13029v1>
- [114] P. Schäfer and U. Leser, “Fast and Accurate Time Series Classification with WEASEL,” *International Conference on Information and Knowledge Management, Proceedings*, vol. Part F131841, pp. 637–646, Jan. 2017, doi: 10.1145/3132847.3132980.
- [115] J. Granderson *et al.*, “Lawrence Berkeley National Laboratory, LBNL FDD Data Sets. DOI: <https://dx.doi.org/10.25984/1881324>,” 2022.
- [116] “. EnergyPlus™. Computer software. Version 00. September 30, 2017. <https://www.osti.gov//servlets/purl/1395882>.”
- [117] X. Fang *et al.*, “Cross temporal-spatial transferability investigation of deep reinforcement learning control strategy in the building HVAC system level,” *Energy*, vol. 263, p. 125679, Jan. 2023, doi: 10.1016/J.ENERGY.2022.125679.
- [118] Q. Zhang, Z. Tian, Y. Lu, J. Niu, and C. Ye, “Experimental study on performance assessments of HVAC cross-domain fault diagnosis methods oriented to incomplete data problems,” *Build Environ*, vol. 236, p. 110264, May 2023, doi: 10.1016/J.BUILDENV.2023.110264.
- [119] C. Fan, W. He, Y. Liu, P. Xue, and Y. Zhao, “A novel image-based transfer learning framework for cross-domain HVAC fault diagnosis: From multi-source data integration to knowledge sharing strategies,” *Energy Build*, vol. 262, p. 111995, May 2022, doi: 10.1016/J.ENBUILD.2022.111995.

- [120] X. Zhu, K. Chen, B. Anduv, X. Jin, and Z. Du, “Transfer learning based methodology for migration and application of fault detection and diagnosis between building chillers for improving energy efficiency,” *Build Environ*, vol. 200, p. 107957, Aug. 2021, doi: 10.1016/J.BUILDENV.2021.107957.
- [121] J. Zhang, Y. Xu, H. Chen, and L. Xing, “A novel building heat pump system semi-supervised fault detection and diagnosis method under small and imbalanced data,” *Eng Appl Artif Intell*, vol. 123, p. 106316, Aug. 2023, doi: 10.1016/J.ENGAPPAI.2023.106316.
- [122] K. Yan, C. Zhong, Z. Ji, and J. Huang, “Semi-supervised learning for early detection and diagnosis of various air handling unit faults,” *Energy Build*, vol. 181, pp. 75–83, Dec. 2018, doi: 10.1016/J.ENBUILD.2018.10.016.
- [123] C. Fan, Y. Liu, X. Liu, Y. Sun, and J. Wang, “A study on semi-supervised learning in enhancing performance of AHU unseen fault detection with limited labeled data,” *Sustain Cities Soc*, vol. 70, p. 102874, Jul. 2021, doi: 10.1016/J.SCS.2021.102874.
- [124] B. Li, F. Cheng, X. Zhang, C. Cui, and W. Cai, “A novel semi-supervised data-driven method for chiller fault diagnosis with unlabeled data,” *Appl Energy*, vol. 285, p. 116459, Mar. 2021, doi: 10.1016/J.APENERGY.2021.116459.
- [125] B. Li, F. Cheng, H. Cai, X. Zhang, and W. Cai, “A semi-supervised approach to fault detection and diagnosis for building HVAC systems based on the modified generative adversarial network,” *Energy Build*, vol. 246, p. 111044, Sep. 2021, doi: 10.1016/J.ENBUILD.2021.111044.
- [126] K. Zhang *et al.*, “Self-Supervised Learning for Time Series Analysis: Taxonomy, Progress, and Prospects,” Jun. 2023, Accessed: Nov. 12, 2023. [Online]. Available: <https://arxiv.org/abs/2306.10125v2>
- [127] A. Vaswani *et al.*, “Attention Is All You Need,” *Adv Neural Inf Process Syst*, vol. 2017-December, pp. 5999–6009, Jun. 2017, Accessed: Nov. 02, 2023. [Online]. Available: <https://arxiv.org/abs/1706.03762v7>
- [128] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *NAACL HLT 2019 - 2019*

- Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, Oct. 2018, Accessed: Nov. 02, 2023. [Online]. Available: <https://arxiv.org/abs/1810.04805v2>
- [129] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *ICLR 2021 - 9th International Conference on Learning Representations*, Oct. 2020, Accessed: Nov. 02, 2023. [Online]. Available: <https://arxiv.org/abs/2010.11929v2>
- [130] D. Wang, X. Wang, and S. Lv, “An Overview of End-to-End Automatic Speech Recognition,” *Symmetry 2019, Vol. 11, Page 1018*, vol. 11, no. 8, p. 1018, Aug. 2019, doi: 10.3390/SYM11081018.
- [131] M. Han, X. Zhang, X. Yuan, J. Jiang, W. Yun, and C. Gao, “A survey on the techniques, applications, and performance of short text semantic similarity,” *Concurr Comput*, vol. 33, no. 5, p. e5971, Mar. 2021, doi: 10.1002/CPE.5971.
- [132] K. Han *et al.*, “A Survey on Vision Transformer,” *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/TPAMI.2022.3152247.
- [133] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, “Efficient Transformers: A Survey,” *ACM Comput Surv*, vol. 55, no. 6, Sep. 2020, doi: 10.1145/3530811.
- [134] P. Dufter, M. Schmitt, and H. Schütze, “Position Information in Transformers: An Overview,” *Computational Linguistics*, vol. 48, no. 3, pp. 733–763, Feb. 2021, doi: 10.1162/coli_a_00445.
- [135] Q. Wen *et al.*, “Transformers in Time Series: A Survey,” *IJCAI International Joint Conference on Artificial Intelligence*, vol. 6, pp. 6778–6786, Aug. 2023, doi: 10.24963/IJCAI.2023/759.
- [136] S. Li *et al.*, “Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting,” *Adv Neural Inf Process Syst*, vol. 32, Jun. 2019, Accessed: Aug. 17, 2024. [Online]. Available: <https://arxiv.org/abs/1907.00235v3>

- [137] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, “A Transformer-based Framework for Multivariate Time Series Representation Learning,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 11, no. 21, pp. 2114–2124, Aug. 2021, doi: 10.1145/3447548.3467401/SUPPL_FILE/A_TRANSFORMERBASED_FRAMEWORK_FOR_MULTIVARIATE-GEORGE_ZERVEAS-SRIDEEDIKA_JAYARAMAN-38957975-XF1A.MP4.
- [138] B. Lim, S. Arik, N. Loeff, and T. Pfister, “Temporal Fusion Transformers for interpretable multi-horizon time series forecasting,” *Int J Forecast*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021, doi: 10.1016/J.IJFORECAST.2021.03.012.
- [139] H. Zhou *et al.*, “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11106–11115, May 2021, doi: 10.1609/AAAI.V35I12.17325.
- [140] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting,” *Adv Neural Inf Process Syst*, vol. 27, pp. 22419–22430, Jun. 2021, Accessed: Aug. 17, 2024. [Online]. Available: <https://arxiv.org/abs/2106.13008v5>
- [141] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, “FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting,” *Proc Mach Learn Res*, vol. 162, pp. 27268–27286, Jan. 2022, Accessed: Aug. 17, 2024. [Online]. Available: <https://arxiv.org/abs/2201.12740v3>
- [142] S. Li *et al.*, “Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting,” *Adv Neural Inf Process Syst*, vol. 32, Jun. 2019, Accessed: Aug. 17, 2024. [Online]. Available: <https://arxiv.org/abs/1907.00235v3>
- [143] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “DeepAR: Probabilistic forecasting with autoregressive recurrent networks,” *Int J Forecast*, vol. 36, no. 3, pp. 1181–1191, Jul. 2020, doi: 10.1016/J.IJFORECAST.2019.07.001.

- [144] J. Xu, H. Wu, J. Wang, and M. Long, “Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy,” *ICLR 2022 - 10th International Conference on Learning Representations*, Oct. 2021, Accessed: Aug. 17, 2024. [Online]. Available: <https://arxiv.org/abs/2110.02642v5>
- [145] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, Dec. 2013, doi: 10.61603/ceas.v2i1.33.
- [146] X. Wang, D. Pi, X. Zhang, H. Liu, and C. Guo, “Variational transformer-based anomaly detection approach for multivariate time series,” *Measurement*, vol. 191, p. 110791, Mar. 2022, doi: 10.1016/J.MEASUREMENT.2022.110791.
- [147] S. Tuli, G. Casale, and N. R. Jennings, “TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data,” *Proceedings of the VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, Jan. 2022, doi: 10.14778/3514061.3514067.
- [148] U. Yokkampon, A. Mowshowitz, S. Chumkamon, and E. Hayashi, “Robust Unsupervised Anomaly Detection With Variational Autoencoder in Multivariate Time Series Data,” *IEEE Access*, vol. 10, pp. 57835–57849, 2022, doi: 10.1109/ACCESS.2022.3178592.
- [149] I. J. Goodfellow *et al.*, “Generative Adversarial Nets,” *Adv Neural Inf Process Syst*, vol. 27, 2014, Accessed: Aug. 17, 2024. [Online]. Available: <http://www.github.com/goodfeli/adversarial>
- [150] S. M. Kazemi *et al.*, “Time2Vec: Learning a Vector Representation of Time,” Jul. 2019, Accessed: Nov. 19, 2023. [Online]. Available: <https://arxiv.org/abs/1907.05321v1>
- [151] A. Siffer, P. A. Fouque, A. Termier, and C. Largouet, “Anomaly detection in streams with extreme value theory,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F129685, pp. 1067–1075, Aug. 2017, doi: 10.1145/3097983.3098144.
- [152] S. Tuli, G. Casale, and N. R. Jennings, “TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data,” Jan. 2022, doi: 10.14778/3514061.3514067.

- [153] S. D. Grimshaw, “Computing maximum likelihood estimates for the generalized pareto distribution,” *Technometrics*, vol. 35, no. 2, pp. 185–191, 1993, doi: 10.1080/00401706.1993.10485040.
- [154] M. Imperadori and F. Brunone, “Active House and user-friendly visualization of sensors’ monitored data: VELUXlab, a real cognitive and smart NZEB prototype,” in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics Publishing, Jul. 2019. doi: 10.1088/1755-1315/296/1/012042.

Appendix

In this Appendix, the investigation in why all the classifiers presented in Chapter 3 performed poorly on the single duct AHU dataset. In Figure 40, visualization of the dataset projected into a 2D plan using PCA. Figure 40 shows that most classes are not easily separable into clusters.

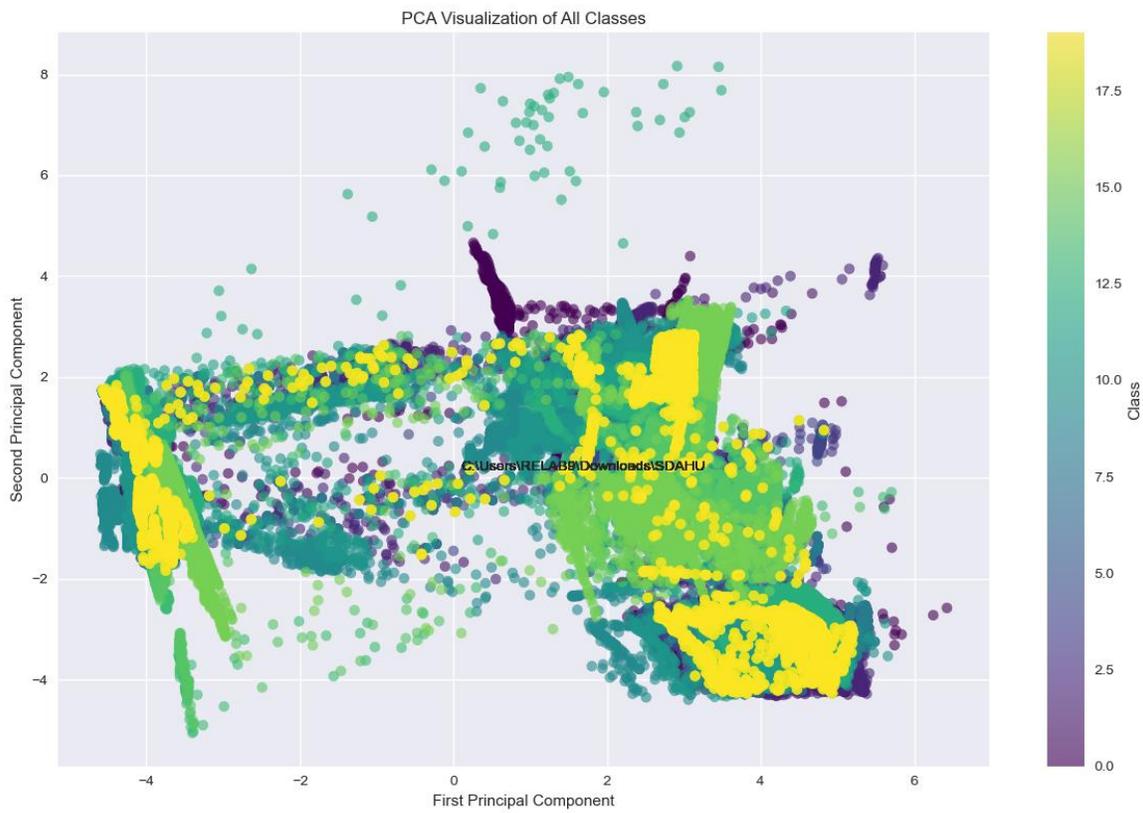


Figure 40: Dimensionality reduction of SD-AHU dataset using PCA.

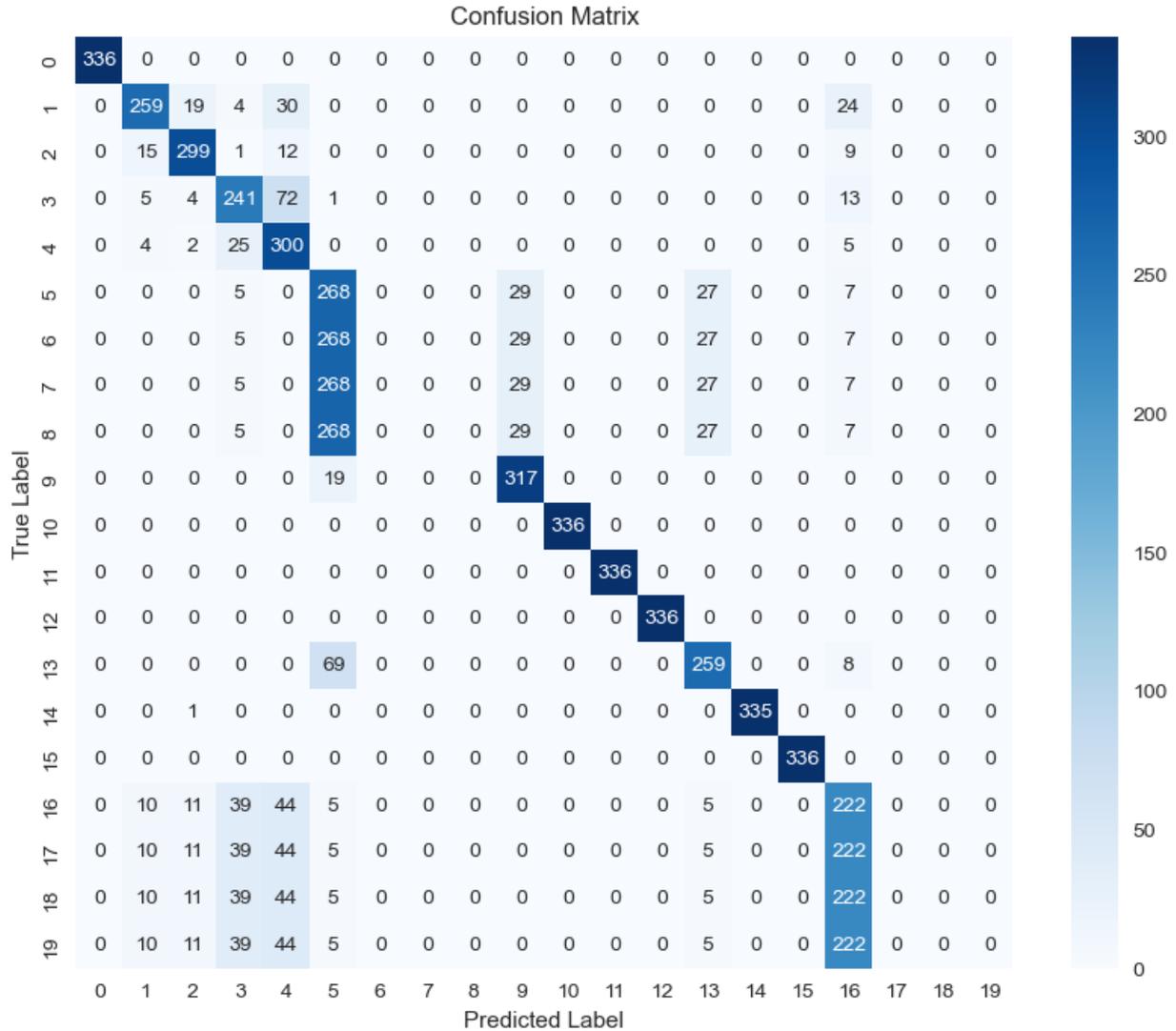


Figure 41 Confusion matrix of LSTMFCNC.

Figure 41 shows the confusion matrix of the LSTMFCNC algorithm. The figure shows some key insights on why the algorithm performed poorly with an accuracy of 0.61. The algorithm is doing relatively well in most of the 20 classes except for 6 classes. Classes 6,7 and 8 are representing the cooling coil valve leakage of 25%, 40% and 50% respectively. The three classes are misclassified as class 5 which is the cooling coil valve leakage of 10% meaning that while the algorithm can detect the fault of the cooling coil valve leakage, it can not distinguish the intensity of that fault. Same case for classes 17, 18 and 19 which represent the outdoor air temperature sensor bias of -4°C, +2°C and +4°C. The algorithm can classify the fault of the sensor bias but can not classify the intensity of the fault.

When plotting the data distribution for the classes 5, 6, 7 and 8, the data distribution is identical for all the files containing the data. This could be either error in the simulation or that the fault intensity does not affect any of the features. The following figures are some of the feature's distribution and time series plots to demonstrate the fact that all the intensities are identical.

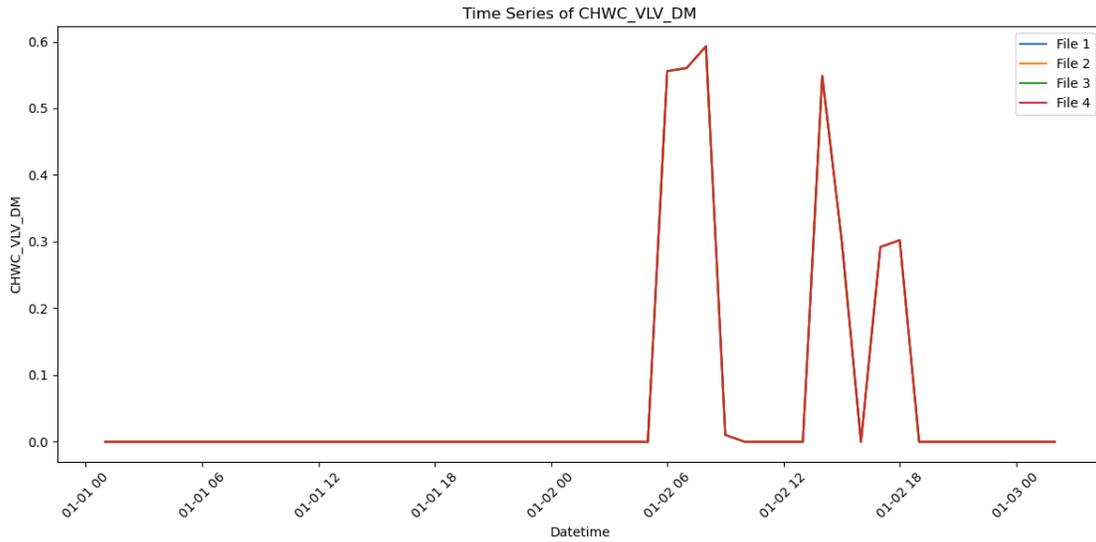


Figure 42 Time series plot of AHU cooling coil valve control signal for the four intensities of the cooling coil valve leakage.

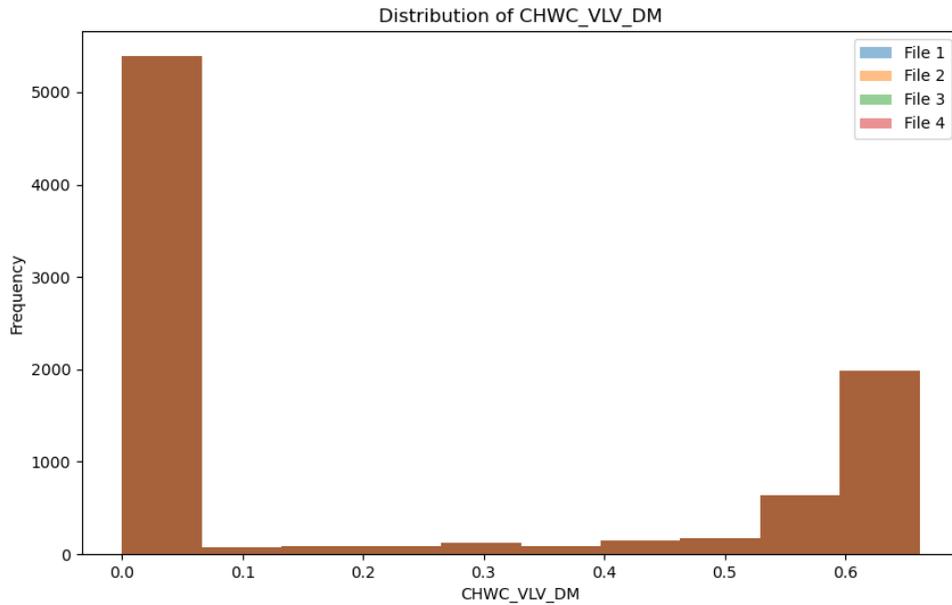


Figure 43 Histogram of the data distribution of AHU cooling coil valve control signal for the four intensities of the cooling coil valve leakage.

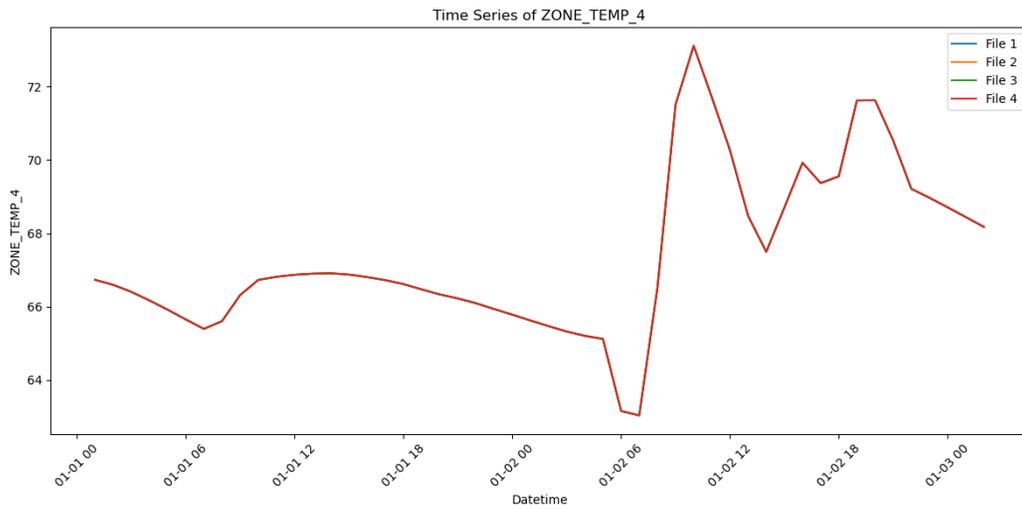


Figure 44 Time series plot of zone4 air temperature for the four intensities of the cooling coil valve leakage.

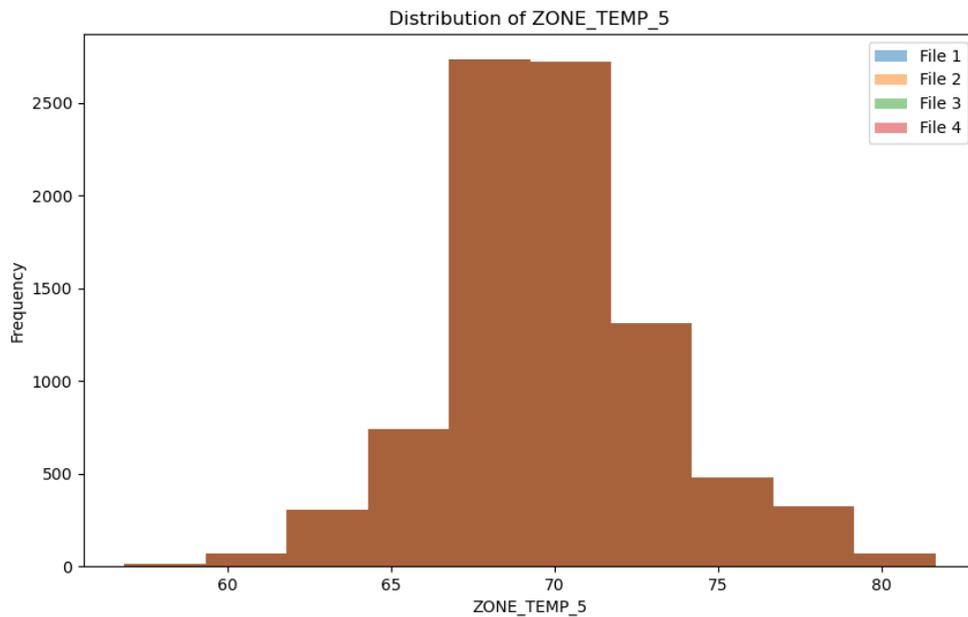


Figure 45 Histogram of zone4 air temperature for the four intensities of the cooling coil valve leakage.

Figure 43 and Figure 45 shows the data distribution of the AHU cooling coil valve control signal and zone 4 air temperature respectively for the four intensities of the cooling coil valve leakage faults. While Figure 42 and Figure 44 shows a plot of 3 days of data for the same fault intensities. Those figures confirm the assumption that the four fault intensities have identical data distributions meaning they are indistinguishable from each other.