Project acronym	B4B
Project full name	Brains for Building's Energy Systems
Grant No	MOOI32004
Project duration	4 year (Starting date May 1, 2021)

Deliverable D1.8a

'Approaches to ML & system-based diagnose and conclusions on the beta design of a software module'

Lead beneficiary	TU Eindhoven
Lead authors	Shobhit Chitkara, Anand Thamban, Wim Zeiler
Publication date	July 1st, 2022
Deliverable Status	FINAL
File name	B4B WP1 D1.08a First overview Machine learning report v1.0 (0) FINAL.docx
Reviewers	Laure Itard (TU Delft), Mirjam Harmelink (TU Delft), Coen Hoogervorst (SPIE)
Contact person	Wim Zeiler w.zeiler@bwk.tue.nl

The project was carried out with a Top Sector Energy subsidy from the Ministry of Economic Affairs and Climate, carried out by the Netherlands Enterprise Agency. The specific subsidy for this project concerns MOOI subsidy round 2020

SUMMARY

The built environment is responsible for nearly 37% of energy consumption and is undergoing a digital transformation. Up to 25% of energy is consumed inefficiently due to inadequate setup and/or incomplete utilisation of available data leading to the so-called performance gap. An efficient fault detection and diagnosis (FDD) strategy using the latest insights of Data Analytics, Artificial Intelligence and Machine Learning for air handling units is key to addressing this gap as they use most energy when the building is well insulated. Even though numerous FDD approaches have been published, real-world applications are far more complex and rarely discussed. This report deals with FDD software module prototyping and integration aspects and discusses its development for air handling units deployed at two case-study buildings in the Netherlands. The design and development of the FDD software module follow a structured process. Firstly, literature research is utilised to narrow the design space and establish a complete use case for developing the FDD tool. Secondly, the developed use case is handled using a data-driven strategy to generate fault symptoms using a state-of-the-art extreme gradient boosting algorithm (XGBoost). The performance was studied by different machine learning regression models, which predict the return water temperature and the cooling coil valve position. The different machine learning algorithms compared for the study include Support Vector Regression, Artificial Neural Network, and eXtreme Gradient Boosting. The data required for the analysis was obtained from fault-introduced experiments conducted in the Kropman Breda living lab office building. What can be concluded from the experiments can also be used in the other case study buildings of the B4B project as it focused on the generic component-based approach.

TABLE OF CONTENTS

ABBREVIATIONS

AFT	Accelerated Failure Time Model
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AFDD	Automated Fault Detection and Diagnosis
APAR	AHU performance assessment rules
ARMAX	Autoregressive Moving Average with Extra Input
BAS	Building Automation System
CART	Classification And Regression Trees
CMS	Continuous Monitoring System
CNN	Convolutional Neural Networks
DBN	Diagnostic Bayesian Network
ERP	Enterprise Resource Planning
FDA	Fischer Discriminant Analysis
FDD	Fault Detection and Diagnosis
GAN	Generative Adversarial Network
GRNN	General Regression Neural Network
HMM	Hidden Markov Models
IoT	Internet of Things
ML	Machine Learning
PCA	Principal Component Analysis
P&ID	Piping and Instrumentation Diagram
PID	Proportional–Integral–Derivative Controller
SAX	Symbolic Aggregate Approximation
SVM	Support Vector Machine
SVR	Support Vector Regression
TS	Time series
XGBoost	Extreme Gradient Boosting
4S3F	Four Symptoms and Three Faults

1 INTRODUCTION

The built environment contributes to about 37% of the total energy consumption in the Netherlands [Marquart & Lange 2017], which amounts to about 679 PJ. With an increasing warming trend witnessed every year, the cooling demand is expected to increase in the European continent [Lhotka et al. 2018]. The HVAC system aims to maintain thermal comfort and the required indoor air quality for human occupation. Research has shown that recommissioning HVAC systems can lead to a 10-20% savings in energy use [Friedman & Piette 2001, Mills 2011].

The goal of the B4B project is to add operational intelligence to buildings. Buildings need “brains” for self-diagnosis and self-optimization to save energy, consider the user and be an active part of the energy system. These brains represent a significant market value due to the impact these “brains” have on energy bills, health and comfort of occupants, operations and maintenance costs and ease of use. Even in the most modern utility buildings, much energy is wasted due to malfunctioning installations and unexpected user behaviour. In many cases, the quality of the indoor environment is insufficient, and the operating costs are high. Smart meters, building management systems, and the Internet of Things allow the collection of large amounts of data. Using this data to reduce energy consumption, increase comfort, respond flexibly to user behaviour and local energy demand and supply, and save on costs for installation maintenance is seen as promising but is underdeveloped and hardly implemented. Real-time analysis and large amounts of data require Machine Learning and Artificial Intelligence. However, current models and algorithms are not yet fast and efficient enough to make buildings "smarter", and the implementation is a cumbersome and time-consuming exercise. Given the complexity, a collaboration of parties throughout the value chain and an open-source approach is a must to achieve scalable and integrated solutions and system innovation in the installation sector.

The project is grouped into four work packages, in which work is executed in an integrated manner on the required development of smart building control. Below is a summary of the work package WP1: WP 1 Self-diagnostic installations for energy efficiency and smart maintenance’’ focuses on developing smart diagnostic systems to reduce energy losses in buildings by continuously identifying faults in the functioning of the building in an automated manner. These diagnostic systems can also be used for optimising performance maintenance planning.

2 LITERATURE ON FAULT DETECTION AND DIAGNOSIS

2.1 General

Over the previous decade, a noticeable increase has been observed in the number of articles published on Fault Detection, and Diagnosis (FDD) approaches focusing on black-box models [Shi & O'Brien 2019]. Zhao et al. [2019] classified the published methods for fault detection and diagnosis. Using this classification, FDD methods can be identified as Data driven-based or Knowledge driven-based. Data-driven-based methods use process-data collected at buildings, whilst Knowledge driven-based based methods rely on domain experts. For example, Rule-based methods use rule sets written using prior knowledge of the system. Through a literature review, 193 published journal and conference articles on FDD methods for HVAC systems are identified [Chitkara 2022]. The FDD techniques for AHUs in these articles are summarised in Figure 1.

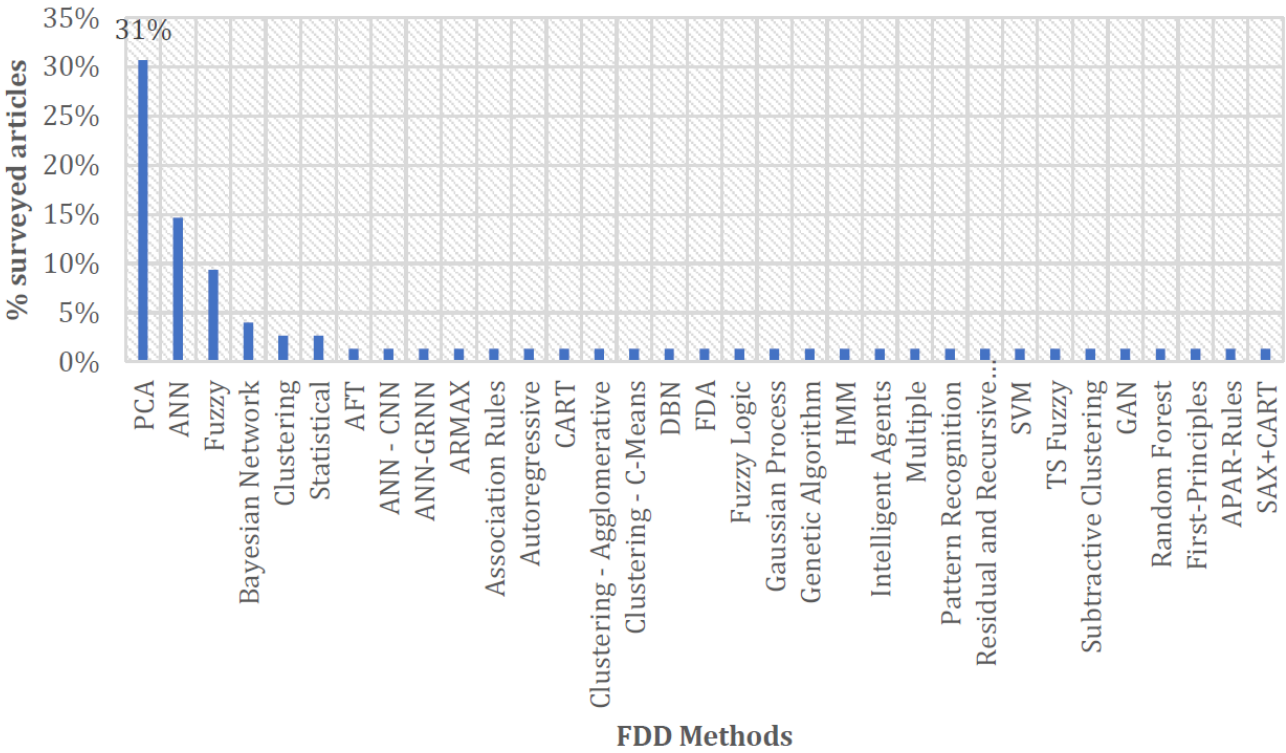

Figure 1: AFDD Techniques for AHU - Ranked in order of popularity [Chitkara 2022]

Labelling data for faulty operation is difficult due to four reasons cited below:

1) The chances of HVAC system operating in the normal state are much higher than the faulty state [Li & O'Neill 2019].
2) It is very expensive and impractical to get sufficient training data for every fault [Fan et al. 2021]. For example, in the RP-1312A project of ASHRAE, it took nearly a year to generate labelled data for 19 different AHU faults. However, it could not cover the wide range of possible operation conditions [Zhao et al. 2019].
3) There is no established or set process for annotation in building operations. Typically, maintenance records or work orders are maintained in Enterprise Resource Planning (ERP) systems and can be read using text mining techniques [Gunay et al. 2018]. However, there is limited interoperability between such systems and FDD or BAS systems. Further, using this approach, it is very difficult to label a particular data point as faulty or normal precisely.
4) Using available measurements at a building, only certain faults can be detected, which makes it difficult to develop models for new or unobserved faults [Fan et al. 2021].

2.2 Fault Detection

Simulation models, 'Fault Modelling', help study fault behaviour inexpensively [Li & O'Neill 2018] and have been utilised for: a) testing FDD methods and b) facilitating fault impact analysis. Fault impact on energy and comfort indicators can be understood using this approach [Zhang & Hong 2017]. A quantifiable indicator such

as fault impact is helpful towards commissioning and prioritisation activities. For fault detection, regression-based or residual generation approaches offer an alternative to working with labelled data. In separate reviews by [Chen et al. 2022] and [Zhao et al. 2019], regression-based approaches constitute 27% and 26% of published research articles, respectively. Here, a black-box model is prepared to model the relationship between inputs (or 'features') and outputs (or 'targets'). Amongst the studied regression-based methods, artificial neural networks (ANN) based and support vector regression (SVR) based approaches are quite popular. Despite this popularity, a comparison between advanced ML techniques made by Chakraborty and Elzarka [2018] and Walker et al. [2019] revealed that gradient-boosting ML algorithms perform better than competing algorithms such as Artificial Neural Networks (ANN), Linear Regression, Random Forest, Support Vector Machine (SVM).

2.3 Fault diagnosis

In their review, Zhao et al. [2019] identified that Bayesian network and Fuzzy logic-based approaches are equally popular. Bayesian network-based approaches have been successfully applied by [Xiao et al. 2014, Zhao et al. 2015, Zhao et al. 2017, Hu et al. 2017, Verbert et al. 2017, Pradham et al. 2021, Najafi et al. 2021] for AHUs. These models generally can handle circumstances when incomplete, uncertain, or conflicting information is presented as their outputs are fault probabilities instead of Boolean fault outcomes [Xiao et al. 2014]. In the context of diagnosis, Bayesian networks are also referred to as Diagnostic Bayesian Networks (DBNs). DBNs are directed acyclic graph models that explain the causal relationships between faults and symptoms. This way, the faults show the possible directions for the diagnoses. The DBN structure is further explained by initial beliefs mapped as prior and conditional probability tables based on expert knowledge [Hu et al. 2018]. These beliefs are updated as new evidence is received using Bayesian inference to compute posterior probabilities. Despite the discussed advantages, the construction of such networks is often tedious and developed diagnosis models lack the interpretability required for widespread adoption. Taal et al. in [Taal et al. 2018], proposed DBNs based 4S3F framework. Here, the 4S implies four generic symptom types, namely energy performance, balance symptoms, operational state, and additional symptoms, and the 3F refers to three different fault categories: model, control, and component faults. Further, this framework draws a clear connection between the developed DBN and the piping and instrumentation (P&I) diagram. Further, its architecture is based on systems engineering theory and how the system can be redistributed across multiple P&I schemes [Taal et al. 2018]. Moreover, the framework proposes a generalisable and automatable approach to creating Bayesian networks for diagnosing faults in HVAC systems. Importantly, the approach has been successfully demonstrated at AHU installations in the Dutch-built environment. Conclusion: Based on all literature, the 4S3F method is the most promising for fault modelling and diagnosis methods.

2.4 FDD Applications

Energy conservation for building and community systems (ECBCS) is a global program piloted by International Energy Agency (IEA). Annex 47 is a sub-track of this program that concerns cost-effective commissioning for existing and low-energy buildings [Ferretti & Choinière 2006]. Within Annex 47, a total of 18 FDD tools were surveyed, and it was identified that a) automation and robust application is highly desirable, b) the developed interfaces of the tool need improvement, and c) they require better integration with the commissioning process [Choinière et al. 2014].

Granderson et al. [2017] surveyed commercially deployed and under-development FDD tools. It can be observed from their survey that FDD tools being utilised by the industry typically rely on a combination of expert rules or first principles. For example, [Bruton et al. 2014] proposed a cloud based FDD tool for AHUs. Their tool utilises AHU performance assessment rules (APAR) [Schein et al. 2006]. Granderson et al. [2021] surveyed 14 commercially deployed tools and noted that whilst their software stack was proprietary several vendors offered application programming interface (API) to support integration.

Prouzeau et al. [2018], pointed out that despite the availability of large sets of data collected through BMS or Internet of Things (IoT) sensors, its visualisation is not effective enough to support building managers for Cx or FDD processes. For applying AFDD for AHUs effectively, identifying the operation modes of AHUs is key [Bruton et al. 2014a]. The bespoke nature of AHU design and its logical operation determined by the control system vendor makes this process quite complicated [Bruton et al. 2014b].

2.5 Conclusions

There have been several FDD applications which were not all too successful. Key characteristics to make FDD applications successful are:

- Good sensor infrastructure
- Use generic Key Performance Indicators
- Use grey box models where physics and data can be combined
- Make sure to clean the data before using
- Analytic functions need to be included to deliver value in FM practice

Nine key issues preventing widespread deployment of FDD systems are listed below {Shobhit 2022]:

1) Rules-based systems heavily rely on sensed information. Due to the sensitivity of building owners to initial project costs, most building installations only have sensors limited to their control functionality [Zhao et al. 2019]. For example, a space involving the transfer of heat, mass, and light with its outdoor environment, occupants, neighbouring spaces, and various building installations is often monitored through just a thermostat [Shi & O'Brien 2019]. Due to the lack of this additional information, it is difficult to develop reliable AFDD models.
2) The lack of standards regarding the quality and positioning of deployed sensors further complicates the FDD process. Machine Learning (ML) models that rely on data collected through these sensors typically grapple with two kinds of uncertainty: a) epistemic and b) aleatory [Hüllermeier & Waegeman 2021]. It is desirable to carefully handle the epistemic (or reducible) uncertainty to apply model-based approaches [Verhelst et al. 2017] successfully
3) There is a lack of a unified framework for developing generic key performance indicators (KPIs) and associated rules for automated fault diagnosis [Taal & Itard 2020].
4) The limits for generating alarms using the rule-based approaches are typically set at a higher threshold than desired to minimise the number of false positives [Kim et al. 2018]. This reduces the ability of an FDD system to detect faults with lower severity and prevent energy waste significantly.
5) Typically, the alarms configured in CM systems are configured once and are not updated continuously. Hence, they fail to detect symptoms that do not breach obvious thresholds [Bruton et al. 2014b].
6) Approaches that utilise black-box models do not inspire confidence in building practitioners as they are not very interpretable [Kim et al. 2018].
7) Approaches that rely on experts for faults are difficult to maintain and scale. This is because building occupancy patterns or service personnel evolve with time, and knowledge transfer is often difficult.
8) Published research methods on novel FDD techniques start with utilising a prepared dataset. However, the practical application of these methods with operational CMS is rarely discussed.
9) Analytical functions in facility management aren't widespread yet [Nehasil et al. 2021]. Commercial alternatives such as Analytics or Software-as-a-service models need to be explored for software procurement that can guarantee value delivery.

3 FDD TOOL REQUIREMENTS

In addition to the requirements identified through stakeholder interactions, key requirements identified through the literature search are presented in this section. These system requirements are identified in two directions: a) application-specific and b) algorithmic requirements. Herein, application-specific requirement implies the requirements constraining the design of the overall FDD tool, whilst algorithmic requirements are identified to constrain the excessive amount of competing for FDD techniques which is of utmost importance. For algorithms, the following have been identified as key to successful implementation and widespread adoption:

- The linkage between AI or data-driven models and the underlying system, such that deployed models are explainable to the user [Taal et al. 2018].
- Overcome the uncertainty within models trained with historical data, which is inherent due to limited information about the historical operation and how well it represents normal behaviour [Taal & Itard 2020].

3.1 AFDD Tool Architecture

The overall system architecture of the AFDD tool based on the literature review and used in this project is presented in Figure 2.

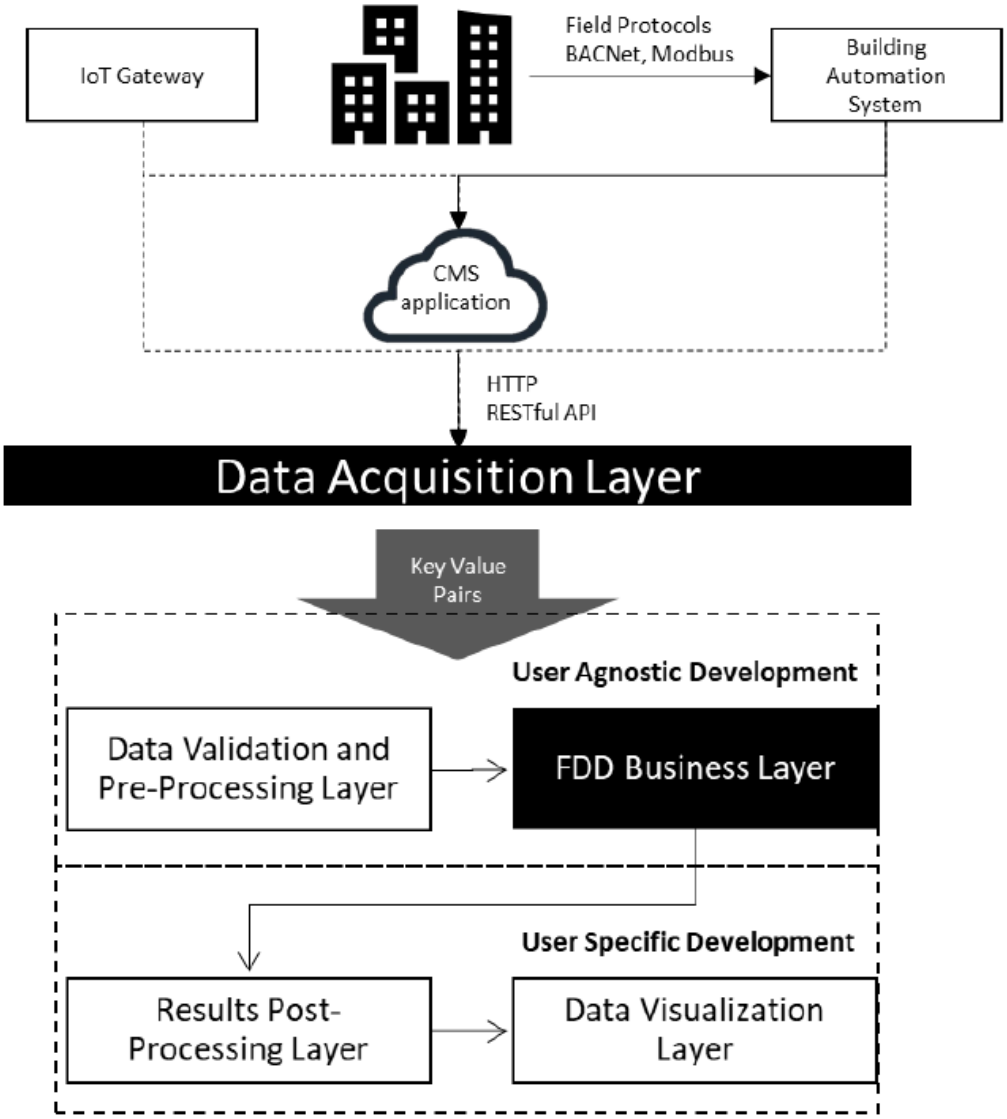

Figure 2: FDD tool prototype architecture

Herein, the implemented workflow is represented with solid arrows. The architecture comprises several layers: data acquisition, data validation and pre-processing business, post-processing, and visualisation. All aspects concerning data transactions with external interfaces, protocols, and security are maintained in the data acquisition layer. The acquired data is then passed into downstream software layers as key-value pairs.

The design of the FDD tool hereafter can be envisioned as User Agnostic and User-Specific development, as shown in Figure 2. The User Agnostic development concerns the design of data preparation and data mining operations on acquired data. These operations are carried out in the Data Validation and Pre-Processing Layer and the FDD Business Layer. The User Specific development is to provide a user-friendly interface for efficiently realising outputs from the FDD business layer and enable human-in-the-loop diagnostics.

3.2 FDD business layer

The FDD business layer has been conceived to wrap critical information such as faults, FDD algorithms, detection thresholds, and fault & symptom associations. It has been designed to keep key domain prioritised functional requirements such as robustness, solubility, and early detection and diagnosis under consideration and prioritised non-functional requirements {Shobhit 2022].

Given these requirements, multiple competing FDD approaches have been studied. Before applying these techniques in practice, a systems analysis step has been introduced (see Figure 3).

Systems analysis supports the realisation of stakeholder needs into definitive product outcomes, and various systems analysis techniques have been proposed to support the product development process across its lifecycle. In this project, systems analysis has been conducted early in its life cycle to a) support planning and development; b) avoid costly design modifications in the latter phases of the AFDD tool development. Two steps in fault and sensor impact analysis formulate this systems analysis step. This approach is also known as the Pareto-LEAN approach [Huls 2016]. However, it is referred to as fault and sensor impact analysis hereafter.

The faults included in the AFDD tool are prioritised using a fault impact analysis. Herein, faults often studied in literature are ranked based on their computed energy gap realised through a simulation approach [Li & O'Neill 2019]. Such prioritisation is vital to maximising the energy saved through the designed tool. A typical large building installation can carry hundreds of sensed and controlled variables. Sensors are prioritised through the sensor impact analysis, and the relationship between faults and analytical symptoms is understood. Using this approach, the design variables involved in the FDD process are constrained.

It has been identified that DBNs can successfully handle the uncertainty associated with the FDD process. The DBN-based 4S3F framework links the underlying HVAC system and the diagnosis process [Taal et al. 2018]. The probabilistic nature of this approach allows a way to eliminate failure modes, which is also synonymous with how HVAC engineers work [Taal et al. 2018]. Further, the relationships between faults and symptoms are characterised using a belief network [Hu et al. 2018]. These belief networks lessen the reliance on the accuracy of the fault detection algorithm, and simplified models can be utilised instead of complex algorithms that offer lesser generalizability [Najafi et al. 2012]. Citing these reasons, the 4S3F framework (see Figure 3) is selected for developing the business layer of the proposed FDD tool.

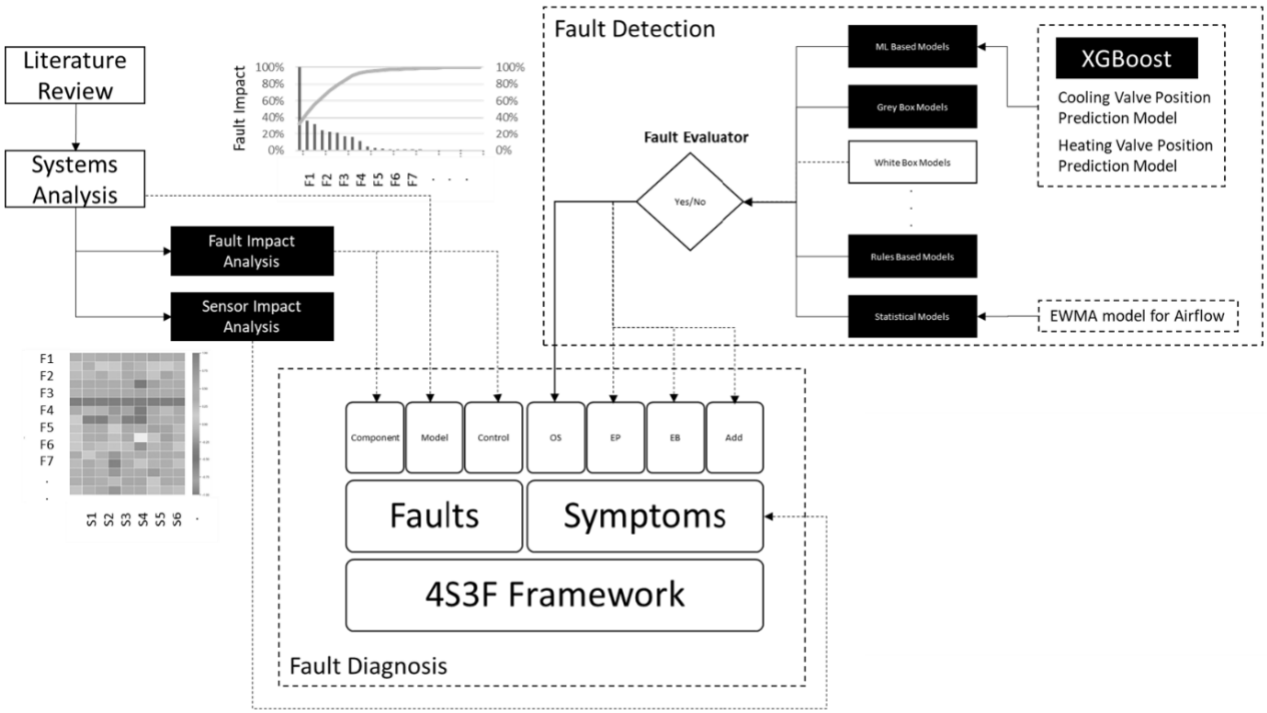

Figure 3: FDD Business Layer

4S3F is categorised under fault diagnosis. However, the whole '4S'part of the framework is about detection. The S stands for 'symptom' and is identical to 'fault detection'. Only 'fault detection' is somewhat misleading as what you detect is not necessarily a fault but a symptom of one or more faults. As the name implies, the 4S3F method classifies faults into three categories: component faults, control faults, and model faults [Taal et al. 2018]:

1) Component faults: Referred to as hard faults that are caused due to design, poor selection, performance degradation or complete failures.
2) Control faults: Also known as soft faults. As the name suggests, these faults refer to issues arising from improper control, such as set points and controller issues.
3) Model faults: These are also called soft faults and are caused because of models deployed for quantitative estimation.

For the faults included in the DBN, the prior beliefs are derived using a literature search [Zhao et al. 2015, Zhao et al. 2017, Wang & Chen 2016] and/or HVAC expert knowledge. The conditional probabilities are prepared using HVAC expert knowledge. In this regard, sensitivity analysis by Taal & Itard [2020] revealed that if the set probability values are reasonable, the likely diagnosed outcome is not affected.

In a DBN, symptom nodes are variables utilised to update prior beliefs on the presence or absence of a fault. In buildings, these nodes are supplied with evidence using data collected from the BMS. It is desirable to utilise smaller Bayesian networks to keep the size of the conditional probability tables manageable; otherwise, the size of these tables grows exponentially [Xiao et al. 2014]. NoisyMax simplification is utilised as a convention for child nodes with more than two parents [Xiao et al. 2014].

Focusing on the symptoms block shown in Figure 3, four kinds of symptom nodes have been proposed by [Taal et al. 2018]

1) Operational state (OS): The OS symptom node represent a deviation in the operational state from its expected state. The OS symptoms can be further classified into control-based and design-based OS indicators [Taal & Itard 2020].
2) Energy Performance (EP): The EP node is representative of normalised key performance indicators such as COP and kWh/m2 that are conventionally utilised to gauge or compare performance within or between systems. They can be further declassified into performance factors, capacity indicators, or energy outliers [Taal & Itard 2020].
3) Energy Balance (EB): The EB node encapsulates system design principles that tend to promote balanced behaviour beyond some transient aberrations. Therefore, fundamental balancing equations are utilised instead of detailed white box modelling approaches [Taal et al. 2018].

4) Additional (Add.): Additional symptom nodes are utilised to pass qualitative or quantitative information from other available information sources such as maintenance logs, manufacturers' input, or user satisfaction [Taal & Itard 2020].

These symptom nodes are activated using fault detection models encapsulated in the Fault Detection Layer (see Figure 3). Diving further into the fault detection layer, it comprises a modelling layer and a fault evaluation layer. For fault detection, amongst the data-driven based approaches, regression-based and statistical modelling approaches have been utilised to set performance benchmarks to distinguish between faulty and expected behaviour. However, the framework is designed with enough flexibility to replace the utilized methods with any competing or superior method given a use-case scenario. This is done to ensure that the developed prototype can be continuously improved with evolution in the AI domain. The fault evaluation is done by setting thresholds on the residual generated through this process. If these thresholds are breached, a fault is detected, and the corresponding symptom is passed to the DBN for diagnosis.

In the FDD Business layer, the fault diagnosis process that concerns root-cause elimination is separated from the fault detection process, which is more aligned with the so-called anomaly detection process. This separation between layers is highly recommended as it allows for multiple techniques from various domains and sub-domains to be combined in a common framework [Shi & O'Brien 2019]. For example, this project uses an advanced AI algorithm called XGBoost (extreme gradient boosting) in the fault detection process, and its outputs are fed into the symptom nodes [Chen & Guestrin 2016].

3.3 AFDD tool integration

Besides the veracity of the business logic, a robust coupling between the AFDD tool and the deployed CMS (InsiteSuite) is key to its usefulness. An application programming interface (API) approach is adopted for integration with this CMS system. This strategy can securely acquire data over standard web protocols such as HTTP. The data acquisition layer has been customised to the available API environment for this project. Although, it can be expanded upon to interface directly with on-premises servers or Internet of Things (IoT) gateways, as shown in Figure 2. This is the first step towards ensuring the desirable interoperability of AFDD application (see Table 3). Further, it ensures that evidence from the BAS deployed on-site is continuously collected and inferred through the utilised FDD strategy.

As the proposed tool utilises AI approaches, the software architecture must attune to this atypical programming environment. Ameisen [2020], discussed how the development of machine learning applications at its core comprises of two pipelines, namely the training pipeline and the inference pipeline. For deploying the discussed XGBoost fault detection model, the two stitched pipelines are shown in Figure 4. The training pipeline starts with data acquisition over the API and ends with a trained model. The performance of the trained model is ensured through the intermediary steps of pre-processing, evaluation through cross-validation, feature selection, and tuning [Leprince et al. 2021]. To ensure multiple modelling approaches, such as (ANN, Gradient Boosting etc.) can utilise the same datasets, the steps until feature selection shown in the training pipeline below are bucketed into the data validation and pre-processing layer (see Figure 2).

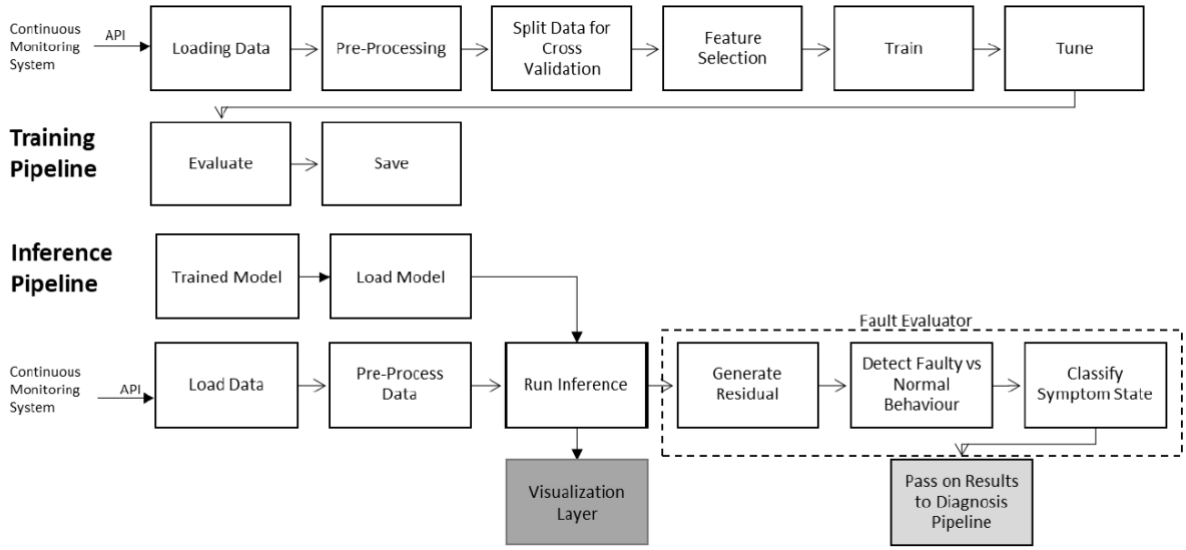

Figure 4: Training and inference pipelines for fault detection

In the inference pipeline, data is requested over the same API and results are inferred using a saved model from the training pipeline. Given the use-case, the inference pipeline splits into two data streams. One is utilised for plotting results from the trained model for visual diagnosis and the other for the fault evaluator. In the fault evaluator, the generated residual is classified as normal or faulty using a fixed or dynamic threshold [Chakraborty & Elzarka 2019]. Hereafter, the results are prepared for treatment in the diagnosis pipeline. For example, the measured continuous variables are converted to discretised as required for the diagnosis pipeline realised using DBNs. The training and inference pipelines for other trained ML models and the Bayesian network are designed using a similar approach.

Besides the treatment of data within the FDD business layer, for effective visualisation, as required in the Data visualisation layer, an intermediary layer is proposed called results post-processing. Herein, the data is aggregated to be presented in charts or tabular visualisation schemes. In effect, the results post-processing layer can support multiple visualisations schemes across the AFDD tool and is therefore designed as a separate component.

3.4 FDD Business Layer set up

The FDD business layer encapsulates core business logic. In other words, it's the brain of the proposed AFDD tool. Several advanced AI methods have been utilized in the FDD business layer. Most demonstrations of these techniques in published literature either utilize prepared datasets or datasets prepared using simulation models. Data collected from real buildings though are prone to uncertainty. Therefore, a set of verification and validation cases have been considered to ensure a reliable development of this layer.

FAULT DETECTION

For fault detection, a reference benchmark is obtained by training a model using historical data collected at the site. The trained model was then used to compare the actual measurement with the predicted value to generate a residual. This process of fault detection is commonly referred to as the regression-based approach [Zhao et al. 2019] and is shown in Figure 5

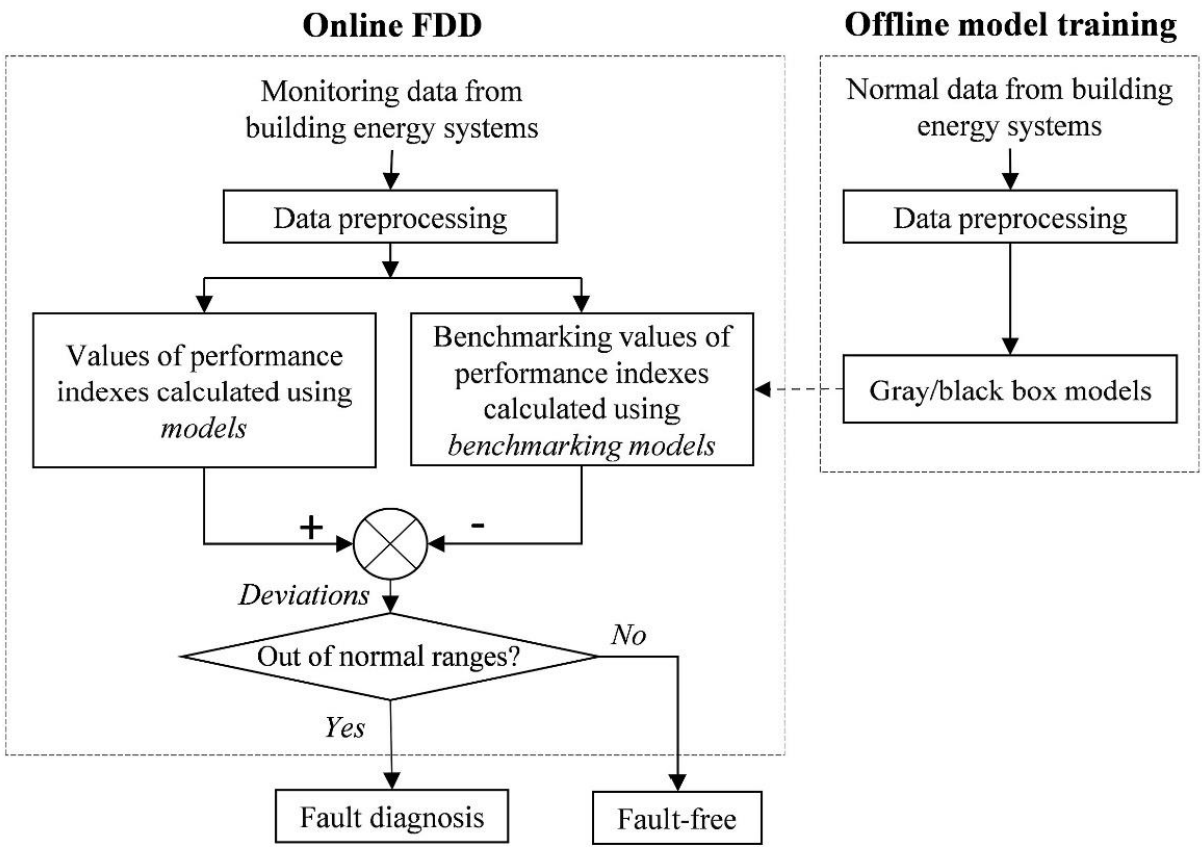

Figure 5: Regression-based fault detection [Zhao et al. 2019]

In this project, XGBoost framework has been utilised in the offline model training process (see Figure 5) given its superior performance [Chakraborty & Elzarka 2018, Walker et al. 2020, Miller et al. 2020]. XGBoost is a scalable, end-to-end machine-learning framework for boosted trees [55]. Its success is attributable to the

popular machine learning theory that multiple weak learners perform better than a single strong learner. In its implementation with decision trees, multiple simultaneous trees are populated much like random forest. Then the predictions from each of these trees are iteratively improved to minimise a loss function such as a squared-error loss or, in other words, mean squared error. The simplified version of the objective function for the algorithm is shown in equation (1) [Chen & Guestrin 2016].

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i] + \Omega(f_t) \quad (1)$$

Here, the first term comprises the first and second-order gradient statistics or the gradient and the hessian of the convex loss function. The second term Ω is the regularisation parameter that penalises the complexity of the model and helps with its generalization process. The XGBoost implementation allows for the objective function to be customised for different loss functions, from its default squared loss error to log-loss or logistic in line with the learning objective. Further, to account for the bias-variance trade-off, multiple hyperparameters have been provided to regularize the complexity of the model. The complete list of hyperparameters is provided in [XGBoost 2011]. Random or grid search methods are typically employed to find the best combination of hyperparameters. Despite their advantages, random or grid search methods are more intuition-driven than model-driven, making hyperparameter tuning less scalable. R. Shi et al. [2021] utilized a sequential model-based optimization (SMBO) that uses Bayesian optimization updates the hyperparameters in an informed manner. This makes the process more generalizable and less resource intensive. Model-driven hyperparameter tuning process is utilized in this project.

For developing an accurate model, feature selection is of paramount importance. Using minimal features is key to preventing the model from overfitting and reducing complexity. To select relevant features for the model, the adopted framework is shown in Figure 6. Besides the available sensed information, new features are engineered. Then, using an iterative process, unimportant features are dropped. The process begins with a coarse feature selection [Zhang et al. 2020]. In this step, multi-collinear features are filtered out using Pearson Correlation Coefficient (PCC) scores. It is followed by a wrapper-based feature selection method known as recursive feature elimination and cross-validation (RFECV). This algorithm recursively eliminates features that carry less impact on the model performance measured using cross-validation [Pedregosa et al. 2011]. The wrapper-based method follows the coarse feature selection process as it is computationally expensive and compensates for the uncertainty associated with the coarse method [Zhang et al. 2020]. The Shapley additive explanations (SHAP) framework is invoked for further dropping features [Lundberg & Lee 2017].

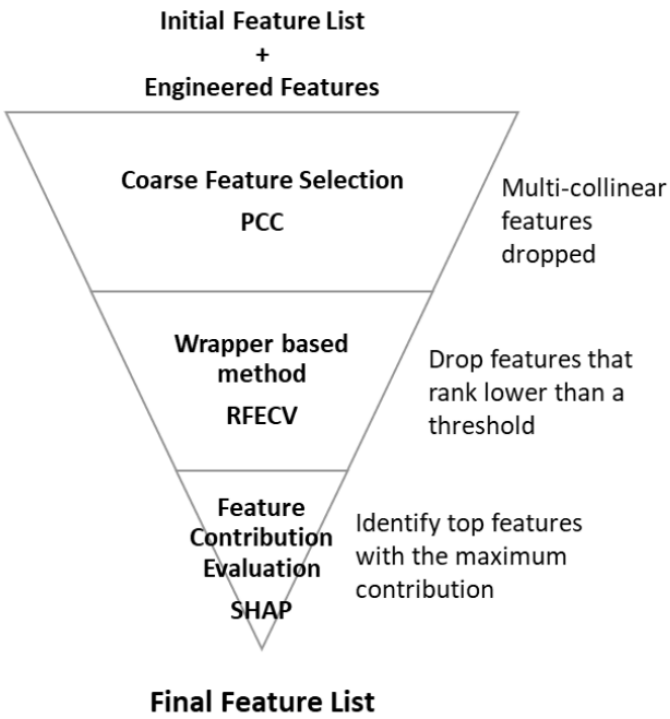

Figure 6: Feature selection framework

Black box models such XGBoost are often difficult to interpret due to the non-linear relationships explored within these models. SHAP is a game theory based explainable AI framework, that helps interpret a trained black-box model. Using SHAP, features that increase model complexity can be removed using an objective evaluation, which improves the model's reliability and generalizability. At its core lie the Shapley values, which average the marginal contribution of all permutations of features/predictors supplied to a model. SHAP is an extension of Shapley values that offers local and global explanations for a predictor. This implies that each prediction made by the model can be individually interpreted, which makes this approach very powerful.

A baseline model is trained using the features selected with RFECV, and multiple candidate models are trained by iteratively dropping features identified as less important using SHAP. Candidate models with a minimal possible but adequate number of features are selected to prevent uncertainty in sensor measurements from carrying into the inference process.

3.5 Product architecture

The FDD tool's architecture consists of the tool's different functional blocks and how each of them interacts. Figure 7 shows the product architecture with the different modules present in it. Each of the modules is explained as different sub-sections below.

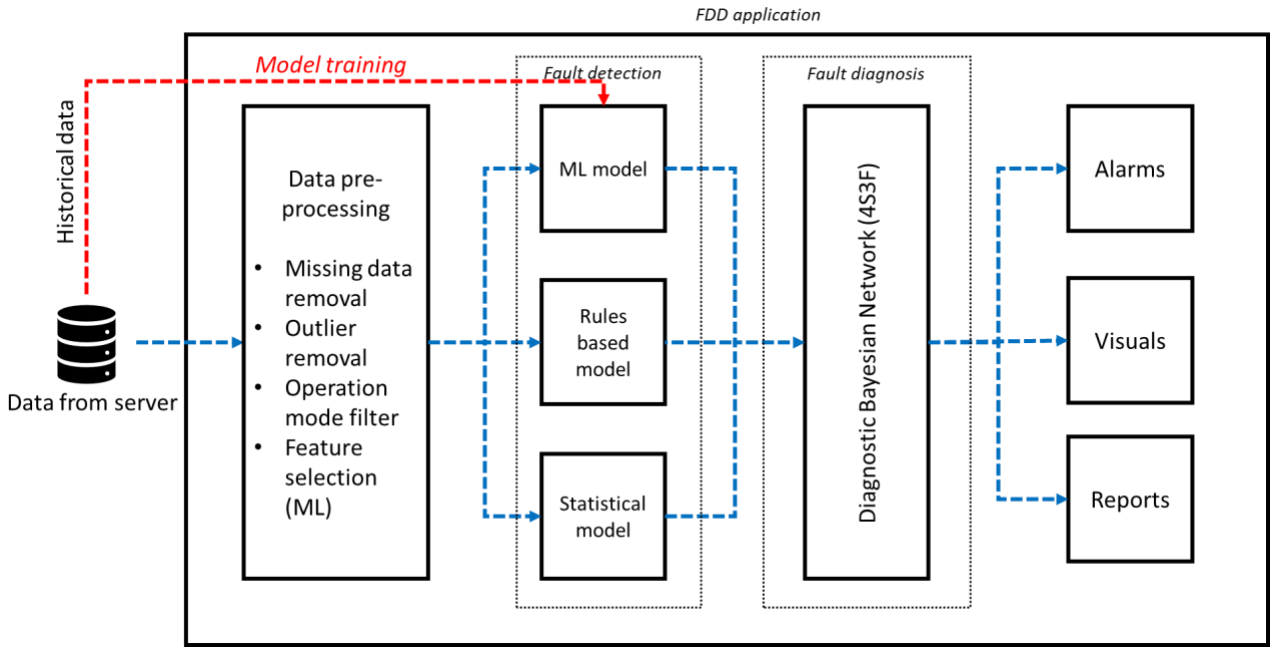

Figure 7: FDD tool product architecture

Data pre-processing

The first step of the FDD process is to clean the data to ensure that noise and outliers do not influence the FDD process. The pre-processing layer consists of missing data removal, outlier removal and operational mode (cooling mode) filters of the AHU. The filtered data is then passed through the fault detection module.

Fault detection

The cleaned data is passed through the fault detection module which consists of different models where certain data points pass through specific models including ML models, statistical model and simple rules-based models where residuals are generated. The ML models are trained and developed using historical data from the server. This process will be done only in the beginning and when model performance drops below certain thresholds. Once the residuals are generated from each fault detection model, the data is sent to the fault diagnosis module.

Fault diagnosis

The fault diagnosis module consists of a diagnostic Bayesian network (DBN) based on the 4S3F method. The DBN consists of symptom nodes, where all the data from the BMS is fed, and it contains fault nodes, where the posterior probability for each fault is calculated. The symptom nodes receive residuals from each fault detection model and are translated to symptom states readable by the DBN. The DBN then calculates posterior probabilities for the different fault nodes present and maps the faults with the underlying symptoms.

Visualisations

All information related to the FDD process is then displayed in a dashboard in the form of alarms, heat maps, bar plots and line plots. Additional visualisations of the DBN are also present, which show the connection between the faults and where they are located in the HVAC installation layout.

3.6 Product development stage

The main aspect of the FDD tool where the user interaction occurs is the dashboard (visualization), where the user interface and user experience aspects need to be considered. The product dashboard was developed based on the agile software development method. The agile development method includes requirements discovery and solutions improvement through the collaborative effort with the end-user. The main purpose of following such a method was to implement practices such as evolutionary development, continual improvement, flexible responses to changes in requirements and a better understanding of the problem to be solved.

To better understand what the different stakeholders expect, an initial design prototype was developed on the prototyping platform called Figma, see Figure 8. The design prototype was made based on the initial requirements which have been laid out.

Figure 8: Agile software development method

The first prototype was developed on Figma, a web-based vector graphics and prototyping tool designers mainly use for building digital products. The first design prototype was shown to the stakeholders to get their feedback and impressions of how the tool would look and what expectations they have regarding features. Their feedback, expectations and suggestions were considered while updating the prototype layout. A second update of the prototype tool was made to get a final confirmation of the dashboard layout.

Once the design was confirmed, the actual front end of the product was developed on Dash, a python-based dashboarding application from the Plotly library. The application (frontend) developed on Dash is rendered in the web browser using an HTML template with CSS and JavaScript elements referenced inside it. The tool's backend includes algorithms developed using the specific libraries required for fault detection (sci-kit learn) and diagnosis (pomegranate). The first prototype of the tool was made and shown to stakeholders so that they can see how the tool works. Their feedback and suggestions were considered, and the product was updated accordingly.

4 APPLICATIONS ON THREE CASE STUDIES

A total of five case studies have been selected for this project to create a wide range of test and validation scenarios. The differentiating aspects for each of the cases are summarised in Table 1. Of the five selected cases, four are from the Netherlands, including a simulation case wherein Dutch weather file has been utilized to have more examples representative of the Dutch built environment. The only foreign example included in the mix is the Energy resource station building. Several researchers worldwide have utilised datasets from this building for ASHRAE's RP-1312 project to demonstrate competing FDD approaches for AHUs [64–66]. These datasets, therefore, provided a consistent baseline for comparison.

Table 1: Case study description

Description	**Case-Studies**				
	#1: 5-Zone Building	**#2: Energy resource station**	**#3: Hoofddorp office**	**#4: Breda office**	**#5: Nijmegen school**
Simulation/Real Case	Simulation	Real	Real	Simulation and Real	Real
Non-Residential Building Type	Office building	Laboratory Facility	Office building	Office building	School
Location	Netherlands	Iowa, USA	Netherlands	Netherlands	Netherlands
Purpose	Systems Analysis, Verification	Verification	Verification	Systems Analysis and Validation	Validation
AHU description	One central AHU supplying to five-zones	Two central AHUs supplying to three zones each	Four central AHUs	One central AHU supplying to three zones (North, South, and Office 105)	Two central AHUs
Fans	CAV	VAV	CAV	CAV	CAV
Coils	1 heating and 1 cooling coil in central AHU and 5 reheat coils	1 heating and 1 cooling coil in central AHU	1 heating and 1 cooling coil in central AHU	1 heating coil in central AHU and 3 cooling coils along supply air path	1 common heating and cooling coil in central AHU
Heat-Recovery	Rotary Heat Exchanger	Air-Side economizer	Rotary Heat Exchanger	Rotary Heat Exchanger	Rotary Heat Exchanger

4.1 Example Valve position prediction model

For fault modelling purposes, the heating and cooling operation of the AHU is modelled with the valve position prediction model for the cooling coil separately. Mass flow rate measurement can be utilized as a key performance indicator for the considered faults of the cooling coil. However, the mass flow rate is not a commonly deployed measurement in AHUs. This also holds for the case studies considered, except Breda office, where a mass flow rate sensor has been deployed as part of this project. To overcome this challenge and keep the FDD strategy generalisable, valve position has been utilised as a proxy for mass flow rate measurement.

Using the discussed XGBoost framework, cooling and heating coil valve prediction models are prepared. For verification of the complete fault detection approach, Energy Resource Station and Hoofddorp office case studies are utilized. The available datasets deployed at these sites have been partitioned to prepare training and test sets. The accuracy of the trained XGBoost models is quantified using the key performance indicators

such as R2 Score, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Cross-Validation Root Mean Squared Error (CV-RMSE) and reported in Table 2 [Afram & Janabi-Sharifi 2014, Walker et al 2020].

Table 2: Results from verification of trained XGBoost models for valve position prediction

Case-Study Building Name	Energy resource station	Hoofddorp office	Energy resource station	Hoofddorp office
Cooling/Heating Valve Position Prediction Model	Cooling Coil Valve Position Prediction	Heating Coil Valve Position Prediction	Heating Coil Valve Position Prediction	Heating Coil Valve Position Prediction
AHU Reference	AHU B	AHU 2	AHU B	AHU 2
Data Sampling frequency	1 min	8 mins	1 min	8 mins
# Training samples	8606	9190	10257	56095
# Test samples	2151	9190	2565	14024
Test Scores	R2 Score: 0.93 RMSE [%]: 2.89 CV-RMSE 0.06	R2 Score: 0.88 RMSE [%]: 6.79 CV-RMSE 0.08	R2 Score: 0.971 RMSE [%]: 0.643 CV-RMSE: 0.033	R2 Score: 0.902 RMSE [%]: 5.252 CV-RMSE: 0.221

As part of the ASHRAE's RP-1312A project, fault experiments were carried out over three seasonal periods between 2007-2008 at an Energy resource station building [Wen 2010, Wen & Shun 2012]. Experiments carried out in the summer of 2007 between August 19th, and September 09th are utilized for verifying the complete fault detection process. Here, two Air-Handling Units (AHU-A&B) were operated simultaneously, wherein AHU-B operated in normal condition whilst faults were introduced in AHU-A. Since both AHUs supply conditioned air to identical thermal zones and are of the same capacity, learned parameters of XGBoost model trained for AHU-B transfer to AHU-A.

The residual between measured and predicted values is utilized to detect faults. Further, a threshold of ±10% on 0-100% scale has been utilized. Since the fault labels are known apriori the detected fault label is compared with the true label and the results are plotted on a confusion matrix shown below in Figure 9. Setting the threshold at a high of ±10% results in specificity exceeding 97%, however, compromising the sensitivity of the detection algorithm. To balance this trade-off, the threshold can be tuned by plotting a Receiver Operator Characteristics (ROC) curve. To this end, a dynamic threshold-setting method has also been explored. The dynamic threshold method showed encouraging results and has been recommended for future improvements over the fixed threshold method utilised in this project.

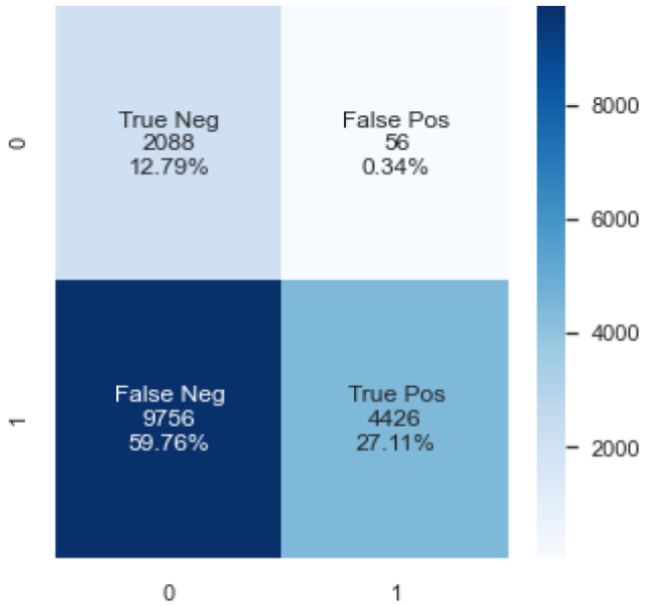

Figure 9: Confusion matrix for fault detection using the developed model

The introduced feature selection framework has been utilized for selecting the final list of features utilized for XGBoost model training. The features selected at each stage of the process are listed in Table 3. The last step of the feature selection process (see Figure 6) using SHAP is presented in Figure 10.

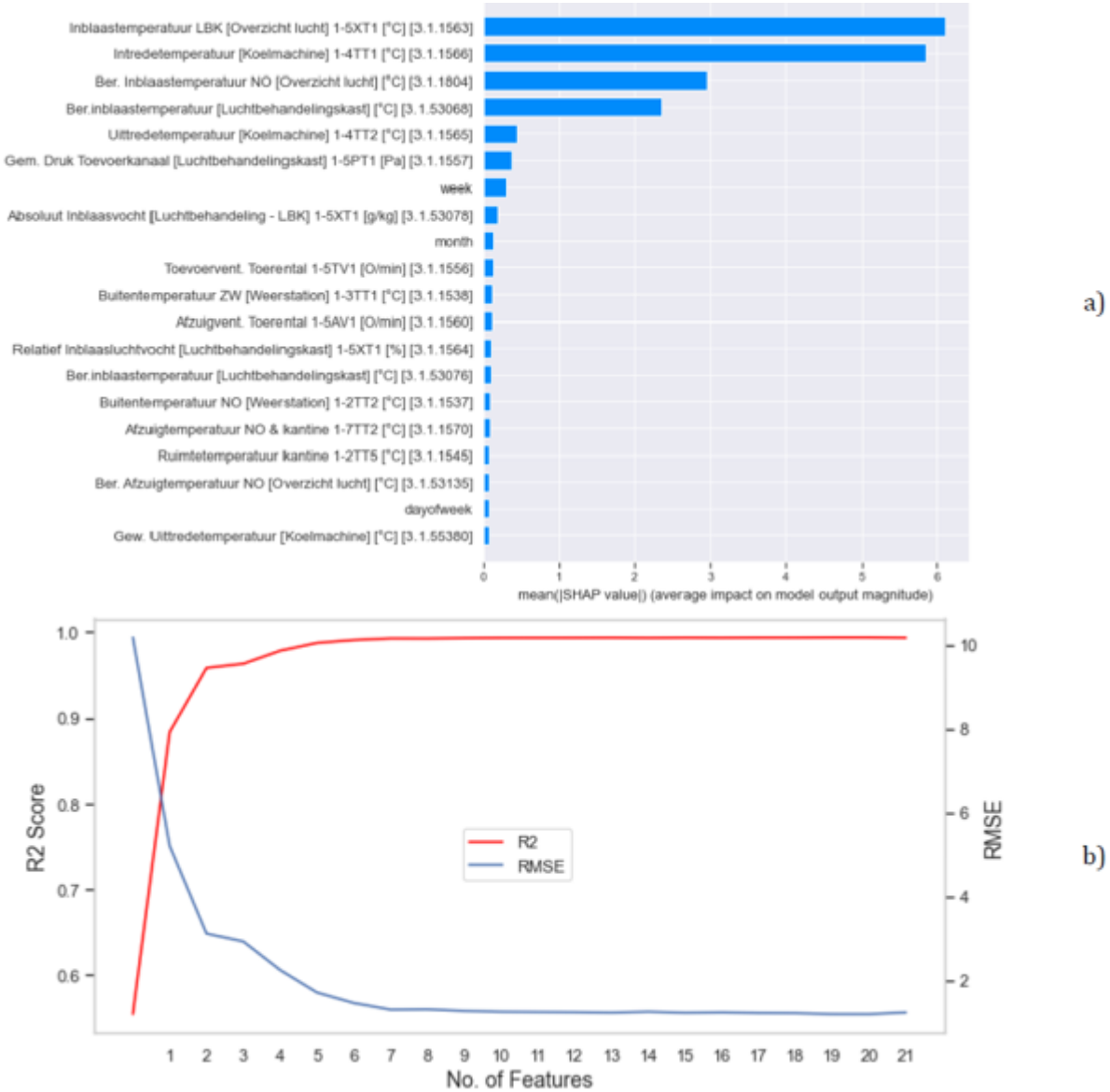

Figure 10: Feature contribution evaluation using SHAP

Table 3: Feature selection for Breda office: North zone cooling coil valve position prediction model

	#Features	Feature Names
AHU Features available from building automation system	19	Outside temperature (North East, South-West), Chiller leaving water temperature, Chiller entering water temperature, Chiller weighted leaving water temperature, Supply air temperature north, Calculated supply air temperature north, Calculated return air temperature north, Return air temperature north and canteen, Supply air temperature central AHU, Room temperature canteen, Relative humidity central AHU, Average static pressure supply air channel, Delta pressure over return air filter, Absolute humidity central AHU supply air, Supply fan speed, Return fan speed, Supply air setpoint,
Engineered Features	5	hour of day, day of week, day of year, week of year, month
Total Features	24	
Features dropped using PCC	1	Supply air temperature
Features Selected Using RFECV	-	None
Feature Selected Using SHAP	7	Supply air temperature (central AHU) Chiller entering water temperature Supply air temperature setpoint (Zone north) Supply air temperature setpoint (central AHU) Chiller leaving water temperature

	#Features	Feature Names
		Supply air pressure Week of year
Final Features	7	Supply air temperature (central AHU) Chiller entering water temperature Supply air temperature setpoint (Zone north) Supply air temperature setpoint (central AHU) Chiller leaving water temperature Supply air pressure Week of year

Since the developed fault detection models are trained with historical data, there is a certain lead time before such models can be deployed. To identify the lead time required for training such models, three cases, namely Hoofddorp office, Breda office and Nijmegen school, have been considered. Here, the models are iteratively trained with datasets partitioned over varying time horizons (a month for Hoofddorp and Breda; weeks for Nijmegen). This variation is synonymous with the length of the available dataset. For example, data from Hoofddorp building is available since 2011, and hence has been partitioned over months. In contrast, data from Nijmegen is only available since March 2020 and has been partitioned in weeks. Starting with a dataset from the first available time horizon, at each iteration, the length of the dataset utilized for training is increased by one. For instance, if the dataset is partitioned in weeks, then the first iteration involves training with dataset from first available week and the second iteration involves training with data from the first week and next available week. This way, it was identified that training an accurate cooling coil valve position prediction model data from at least 20 weeks or nearly a complete cooling season in the Netherlands would be ideal. The results detailing these experiments are presented in Figure 11 and Figure 12.

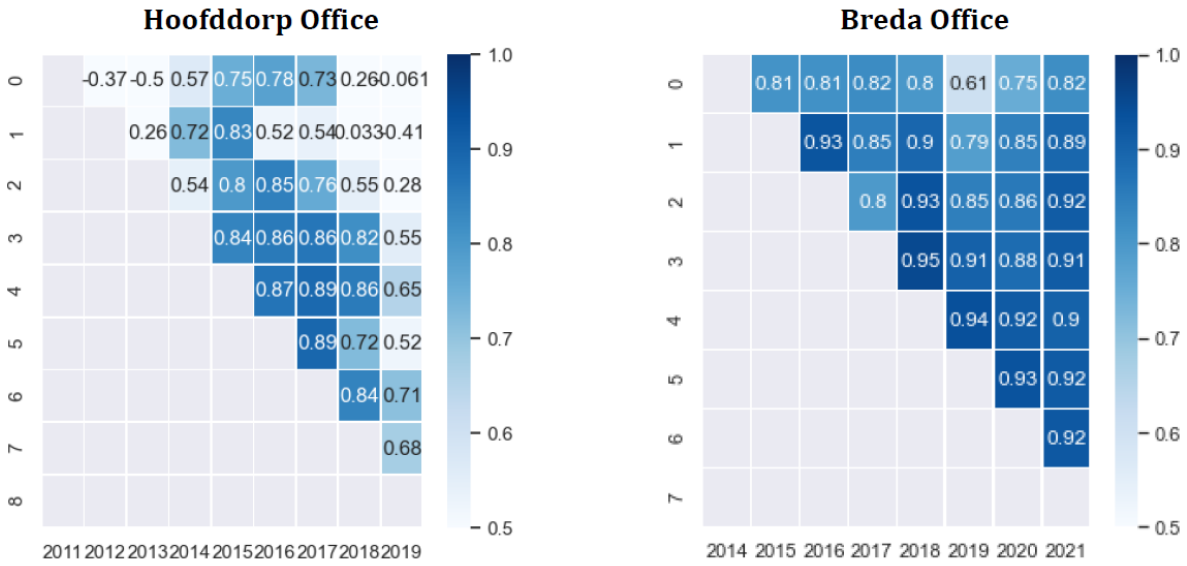

Figure 11: Experiments to evaluate lead-time for black-box models with datasets from Hoofddorp office and Breda office

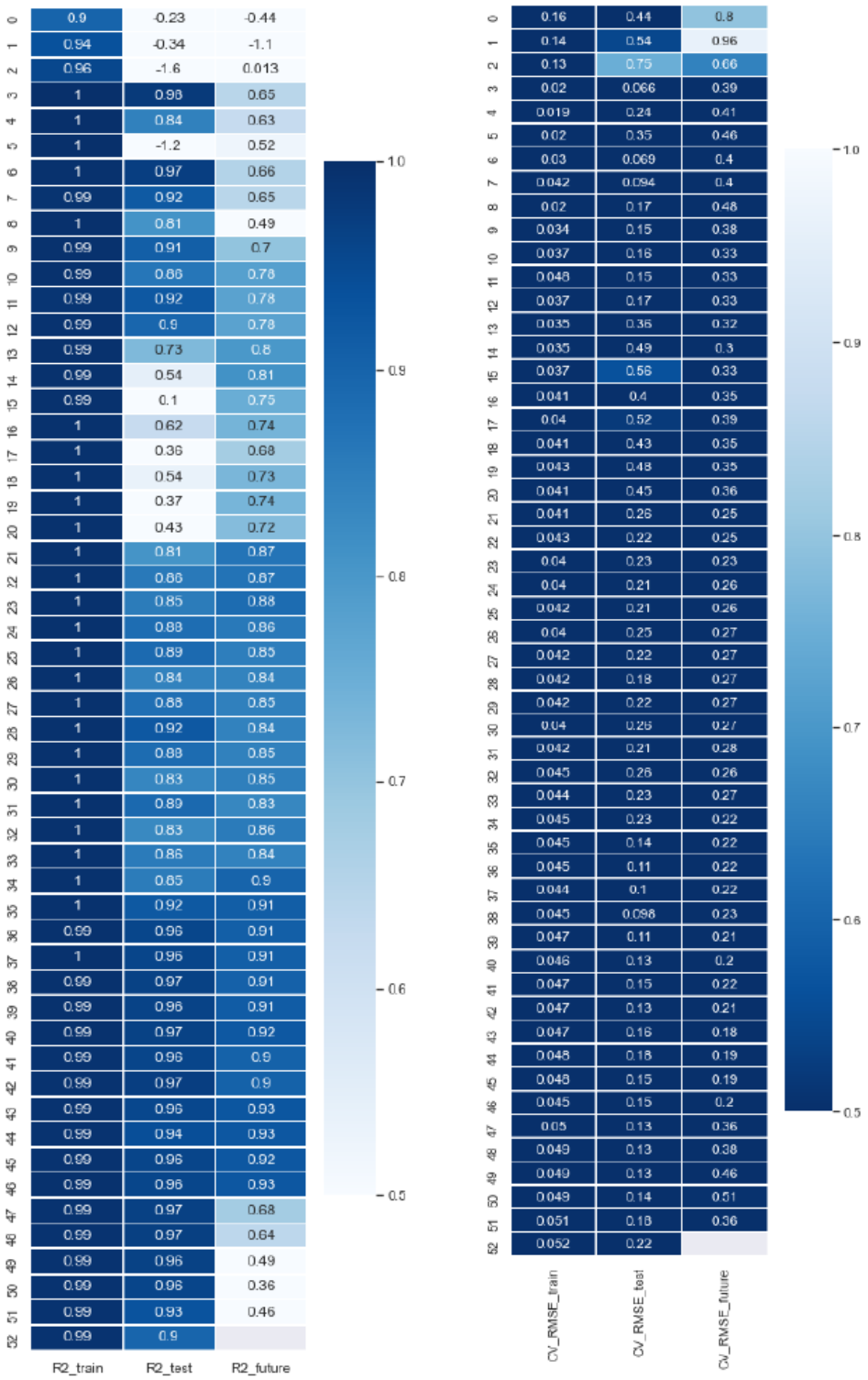

Figure 12: R2 Score and RMSE with weekly tests – ROC Nijmegen

Using the discussed XGBoost algorithm, cooling and heating valve position models are trained for Breda office and Nijmegen school. The scores of these models on the performance indicators discussed previously are tabulated in Table 4.

Table 4: Accuracy metrics for the trained ML models for heating and cooling valve prediction

Model	Site	AHU	Key Performance Indicator	Score
Cooling Coil Valve Position	Breda office	North Zone Cooling Coil	R2 Score	0.94
			MAE [%]	2.63
			RMSE [%]	4.86
			CV-RMSE	0.11
	Nijmegen school	AHU One	R2 Score	0.98
			MAE [%]	2.65
			RMSE [%]	4.25
			CV-RMSE	0.09
Heating Coil Valve Position	Breda office	Central AHU Heating Coil	R2 Score	0.99
			MAE [%]	0.98
			RMSE [%]	2.18
			CV-RMSE	0.05
	Nijmegen school	AHU One	R2 Score	0.99
			MAE [%]	0.91
			RMSE [%]	1.72
			CV-RMSE	0.02

Between the heating and cooling valve position prediction models, it can be observed that heating coil valve position prediction models have better performance scores. Given this better fit and higher accuracy, these models can be utilised with lower thresholds for fault detection and hence are useful for identifying faults with lower severity. On the CV-RMSE indicator, the scores of all trained models are less than 0.3, it can be said that they are useful for engineering applications and are proceeding with further for fault diagnosis [Walker et al 2020].

Using data-driven methods, anomaly detection is a popular way of detecting faults in the data by identifying unexpected or abnormal data from normal fault-free data. This is usually done using supervised learning regression models, which predict fault-free data and then compare with the measured data. When large residuals are identified between the expected value (fault-free) and the measured value (faulty), it can be assumed that a fault exists in the system. Different kinds of machine learning (ML) algorithms have been used for regression purposes, including Artificial Neural Networks (ANN), Support Vector Machines (SVM) or Support Vector Regression (SVR), Decision Trees, Random Forest, and eXtreme Gradient Boosting (XGBoost). Each of these algorithms is advantageous depending on the dataset's characteristics.

SVM or SVR is an algorithm used for both classifications and regression problems. It has the advantage of performing well with a limited amount of data compared to other models. But the computational time required for model development is considerably higher than other ML algorithms like ANN and Random Forest [Walker et al. 2020]. Decision trees are also regression algorithms that are based on the approach of splitting a dataset while evaluating certain conditions. Ensemble algorithms are based on the ML theory that a group of weak learners creates a much stronger ensemble than a single strong learner [Zhou 2021]. XGBoost is one such ensemble algorithm that has proven to be a well-performing ML algorithm in several studies [Pan 2018, Mo et al. 2019, Yao et al. 2019] and has been previously used for fault detection in HVAC systems [Chakraborty & Elzarka 2019]. Since previous research clearly showed the benefits of ensemble algorithms compared to individual Decision Trees [Zhou 2021], Decision Trees are not included in this study. ANN has also been used to develop regression models to predict continuous variables like energy consumption [Walker et al 2022], temperature [Montazeri & Karger 2020] and cooling coil valve position [Wang & Jiang 2004]. But ANN is more complex in nature compared to SVR and requires precise adjustment of its many hyper-parameters [Seyedzadeh et al. 2018]. Neural networks also perform better with larger amounts of data, which could be a drawback if limited data is available [Seyedzadeh et al 2018].

5 COMPARISON OF TWO METHODS

5.1 Machine learning – Performance comparison of Artificial Neural Networks (ANN) and Extreme Gradient Boost (XGBoost)

Since it is very rare to have mass flow rate sensors in installations, a convenient method is to observe the cooling coil valve (CCV) position. ML models can predict the CCV position and the ΔT across the coil. When these predicted values are compared with the actual values and residuals are generated, large residuals (greater than a threshold) would signify the presence of low ΔT syndrome. In contrast, small residuals (smaller than a threshold) indicate normal operating conditions.

Many kinds of ML models can be used, and each can have varying performances based on the quality of the dataset. The ML study for CCV position prediction using SVR showed that the model performed reasonably well (error <15%) for a limited amount of training data. But this would not be the ideal approach to fault detection since drastic weather changes can occur over a short period, and therefore short training data sets (of 1 week) would perform badly. Therefore, it was important to use ML models, which perform much better with larger training data sets. The performance of ANN and XGBoost was compared for the prediction of CCV position and ΔT for two use-case buildings – an office building in Hoofddorp and office building in Breda. In the following subsections, each model is analysed and compared for the different use case buildings.

The features used in these ML models most commonly include outdoor air temperature, the pressure difference across the ducts, fan speed, humidity, cooling coil supply water temp, exhaust air temperature and humidity etc. It is important to avoid all features which can be influenced by certain faults (e.g., the supply air temp after the cooling coil is influenced by both ΔT and cooling coil valve position).

5.1.1 Results using ANN

The office building in Hoofddorp was previously used for the initial ML study using SVR. The unique feature of this building is that there are large amounts of data available for studies. The large dataset (8 min interval data from 2014 – 2020) is ideal for studying the performance of ANN models since ANN works well with a large amount of data. It is to be noted that some data for the year 2017 is unavailable due to a malfunction in one of the sensors.

An important part of model development is to select the right features to obtain the best prediction. In the case of ANNs, there is no automated feature selection/ranking method like Recursive Feature Elimination using Cross-validation (RFECV) which is used in XGBoost and SVR. Therefore, feature selection is done manually which is not a very desirable trait, especially when the FDD tool is required to be commercially deployed. The hyperparameter tuning of the model is done using grid-search which is called from the scikit-learn library available in python. This part of the model development is common between both ANN and XGBoost and is automated.

Figure 11 shows the performance of the ANN model used in the prediction of CCV position for the office building in Hoofddorp. The model has an R^2 score of 0.86 and a RMSE of 12.5%. It is also observed that the model does not always correctly predict values for instances when the actual CCV position is 100%. These are most likely instances where there is transient behaviour in the system (when the system starts up or shuts down) and therefore a steady-state detector and filter would be an essential pre-processing tool in the FDD tool.

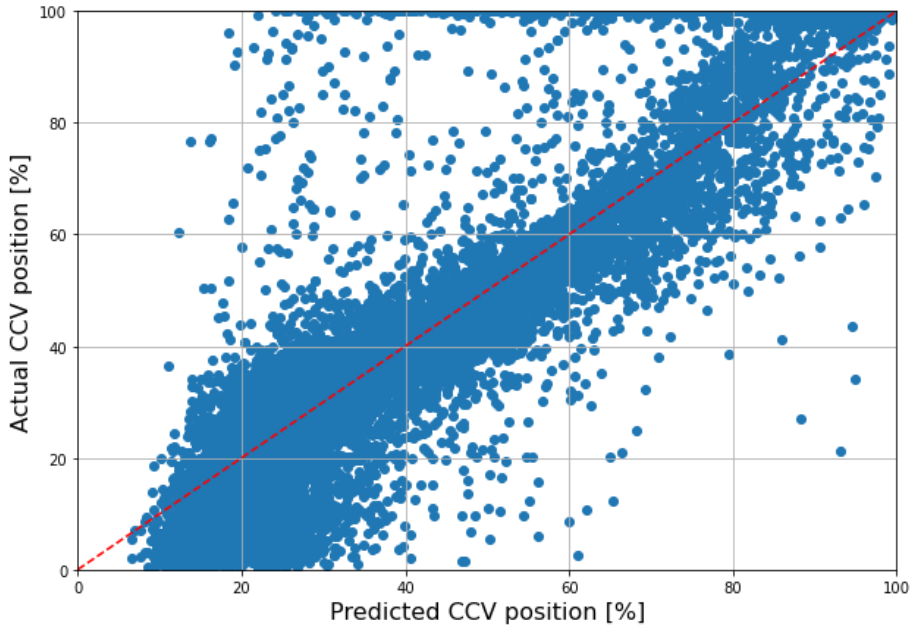

Figure 11: Comparison of predicted and actual CCV position [%] using ANN – building Hoofddorp

5.1.2 Results using XGBoost

Developing the CCV position prediction model using XGBoost was easier since RFECV could be used for feature selection. Along with the automated hyperparameter tuning, it proved to be a much easier model to train and develop. Figure 12 shows the performance of the XGBoost model used in the prediction of CCV position for the office building in Hoofddorp. The model has an R^2 score of 0.89 and RMSE of 11.5%.

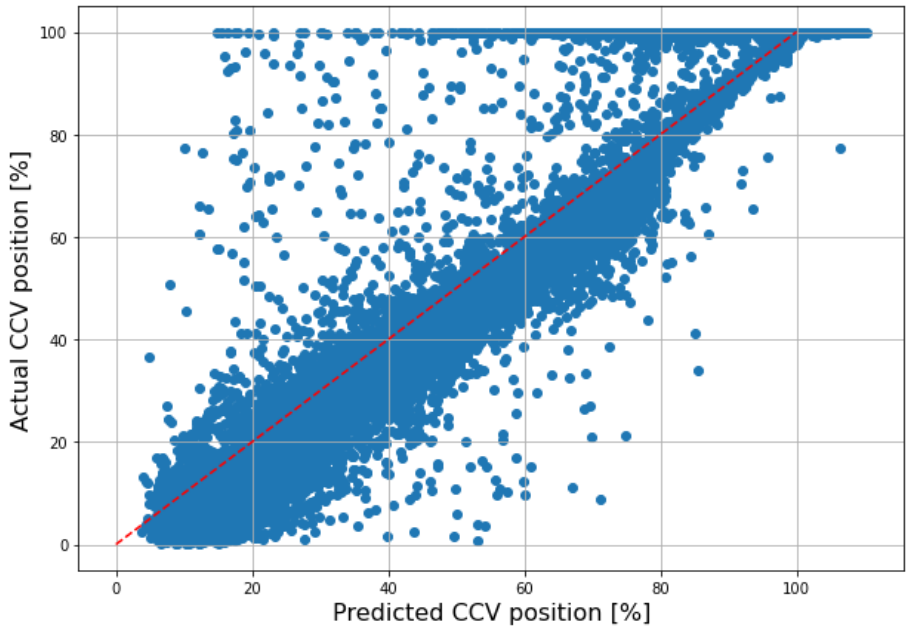

Figure 12: Comparison of predicted and actual CCV position [%] using XGBoost – building Hoofddorp

Both ANN and XGBoost have similar performance in terms of accuracy and RMSE but only differ in terms of the model development process, i.e., automated feature selection.

5.2 ΔT prediction – office building Hoofddorp

5.2.1 Results using ANN

Like the CCV prediction model, a ΔT prediction model was developed using ANN. As mentioned earlier, manual feature selection was done along with grid search for hyperparameter tuning. The resulting model had an R^2 score of 0.88 and RMSE of 0.66K. Figure 13 shows the comparison of the actual ΔT and the predicted ΔT and it is evident that the predicted values are quite close to the trend line.

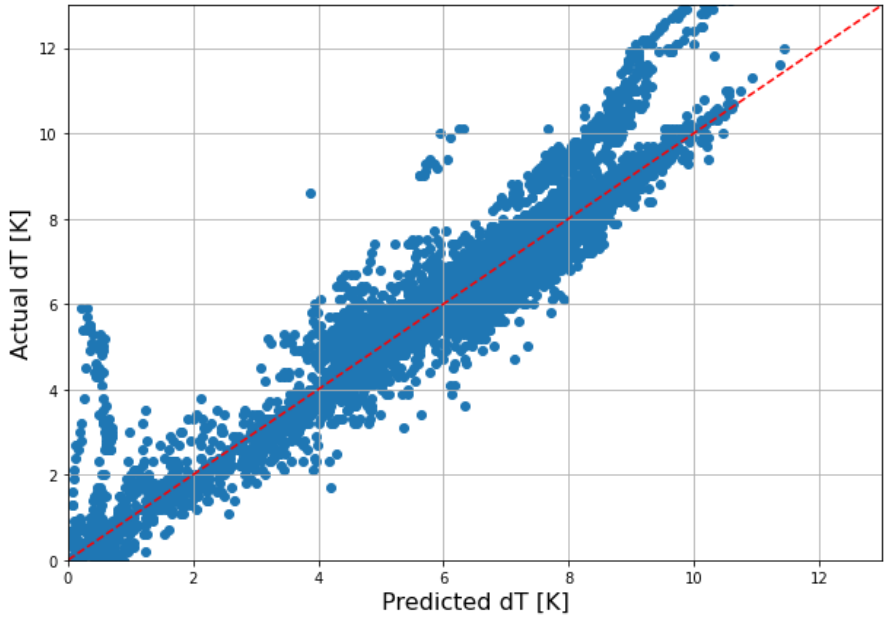

Figure 13: Comparison of predicted and actual ΔT [K] using ANN – building Hoofddorp

5.2.2 Results using XGBoost

The development of the ΔT prediction model using XGBoost was easier since RFECV could be used for feature selection. Along with the automated hyperparameter tuning, it proved to be a much easier model to train and develop. The resulting model had an R^2 score of 0.88 and an RMSE of 0.65K. Figure 14 shows the comparison of the actual ΔT and the predicted ΔT, and it is evident that the predicted values are quite close to the trend line, similar to that observed in the ANN model.

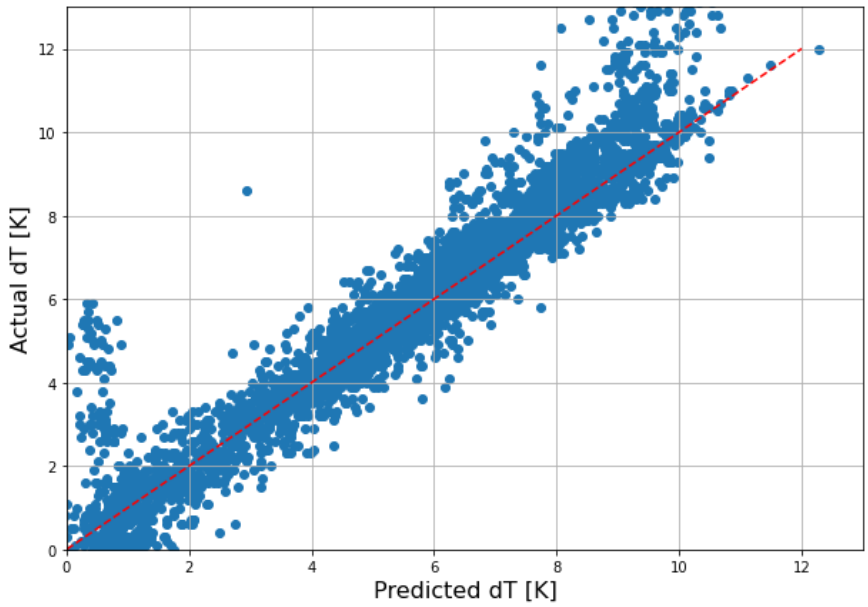

Figure 14: Comparison of predicted and actual ΔT [K] using XGBoost – building Hoofddorp

Both models showed similar performance in terms of R^2 score and RMSE. The next step was to compare its performance based on the size of the training data. 7 shows the R^2 scores and RMSEs for different training data sizes ranging from 1 year to 5 years for the ANN model. It is important to note that the dataset is not arranged randomly and is trained in chronological order of time. Therefore, 3 years of data would mean training between 2014-2016 and testing for 2018 (Since data for 2017 is not available due to sensor failure).

It is observed that the performance of the ANN is model reduced with smaller training data set, and therefore, for optimal performance of an ANN, it is important to have a large dataset for training.

Table 5: Performance comparison based on years of training data using ANN – building Hoofddorp

Years of training data	1	2	3	4	5
R^2 score$_{test}$	0.37	0.75	0.85	0.79	0.88
RMSE$_{test}$	1.5 K	0.943 K	0.72 K	0.85 K	0.66 K

XGBoost on the other hand, did not require a very large training dataset and in fact, gave a reasonable performance with just one year of training data. This is seen in 8.

Table 6: Performance comparison based on years of training data using XGBoost – building Hoofddorp

Years of training data	1	2	3	4	5
R^2 score$_{test}$	0.82	0.85	0.84	0.84	0.88
RMSE$_{test}$	0.793 K	0.73 K	0.75 K	0.76 K	0.66 K

In some situations, very limited data will be available for training (less than 1 year). The office building in Breda is a good example of where limited training data is available. The performance of both XGBoost and ANN for the detection of ΔT are analysed for a very short training set as well.

5.3 Cooling coil valve position prediction – office building Breda

The office building in Breda is a living lab and is specially designed to conduct experiments in the AHU. The setup consists of multiple sensors around the cooling coil to study the low dT syndrome in more detail. The building consists of three target zones for conditioning – North sector, South sector and the main office space (1.05), each of them having individual cooling coils for conditioning the air of the respective zones. In this study, the cooling coil targeted for room 1.05 is studied. The ML model development process is the same as in the previous cases and therefore the results are directly reported.

5.3.1 Results using ANN

Figure 15 shows the comparison of the predicted and actual CCV position values. The generated model had an R^2 score of 0.84 with an RMSE of 11%.

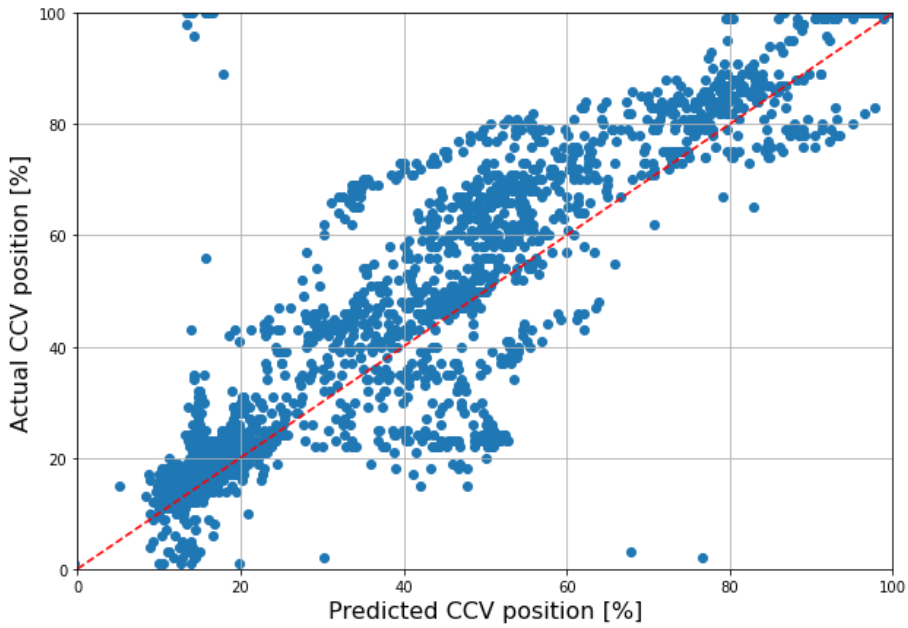

Figure 15: Comparison of predicted and actual CCV position using ANN – building Breda

5.3.2 Results using XGB

The XGB, on the other hand performed slightly better with a model R^2 score of 0.85 and RMSE of 8.7%. Figure 16 shows the comparison of the predicted and actual CCV position values. From both figures, it can be said that both ANN and XGB have similar performance with a sufficient amount of training data.

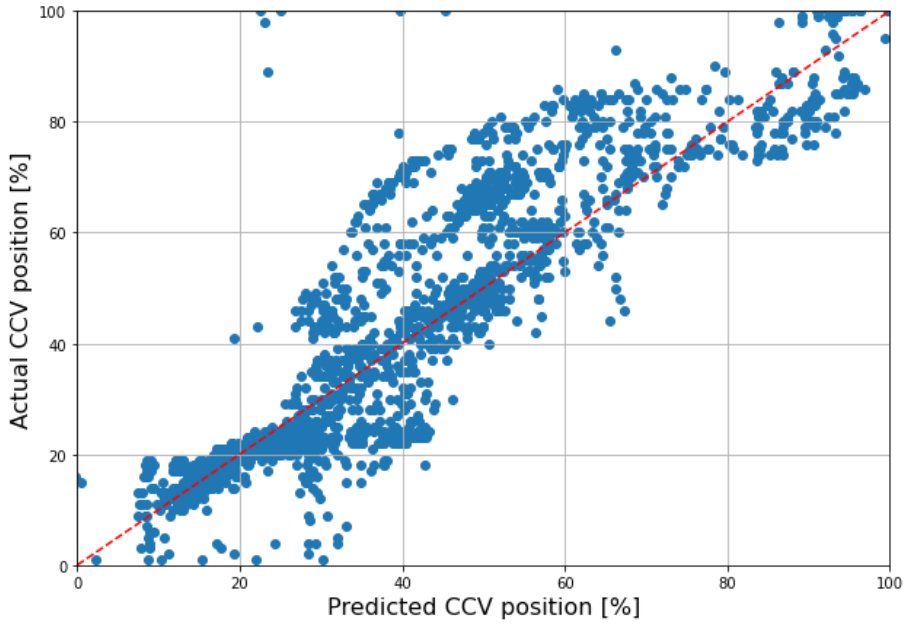

Figure 16: Comparison of predicted and actual CCV position using XGBoost – building Breda

5.4 ΔT prediction – office building Breda

The office building in Breda consists of three target zones for conditioning – North sector, South sector and the main office space (1.05), each having individual cooling coils for cooling the respective zones. In this study, the cooling coil targeted for room 1.05 is studied. The ML model development process is the same as in the previous cases, so the results are directly reported. The ANN model has an R^2 score of 0.47 and an RMSE of 1.15K. Figure 17 shows the comparison of predicted and actual values of ΔT.

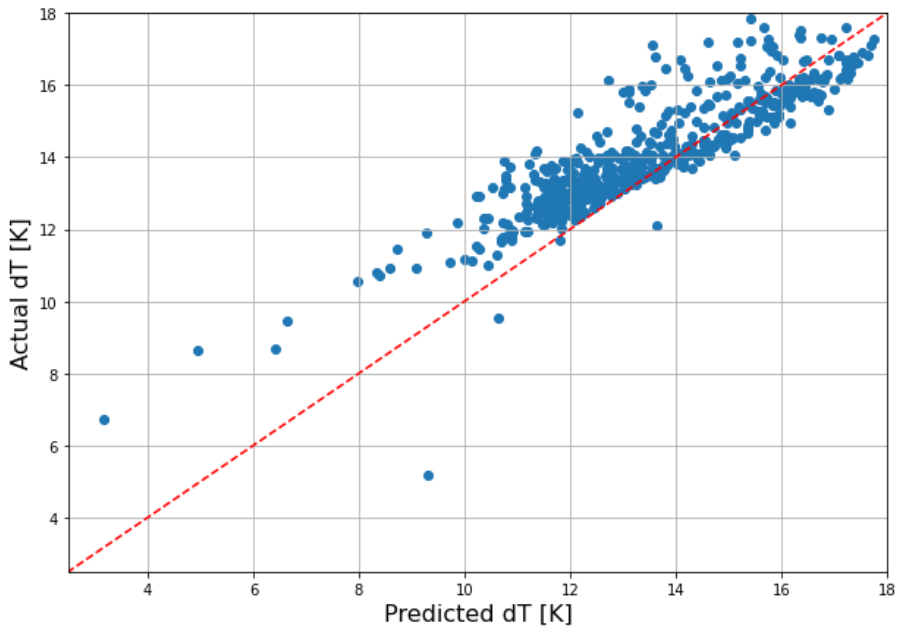

Figure 17: Comparison of predicted and actual ΔT using ANN – building Breda

The XGBoost model has an R^2 score of 0.93, and an RMSE of 0.4 K. Figure 20 shows the comparison of predicted and actual values of ΔT.

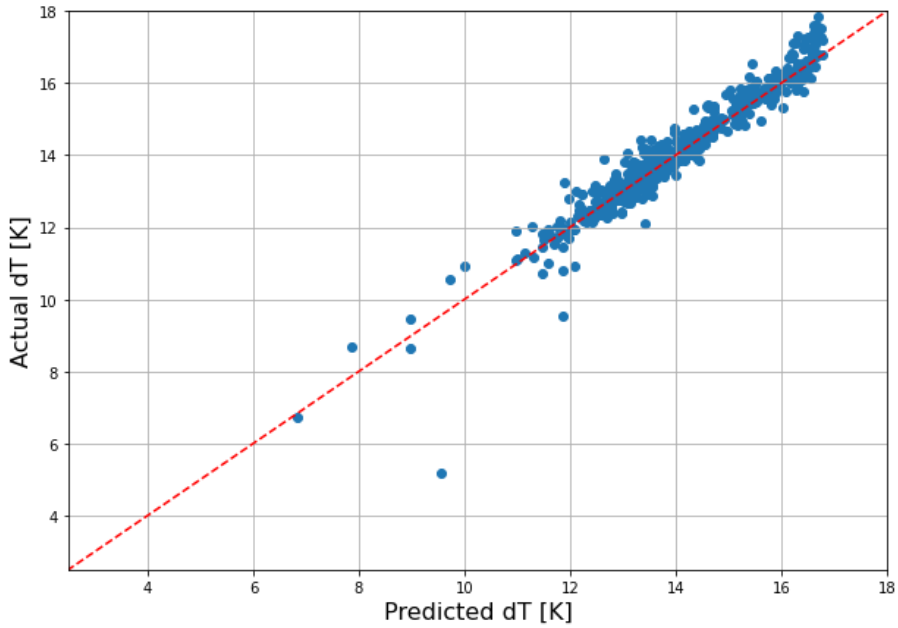

Figure 20: Comparison of predicted and actual ΔT using XGBoost – building Breda

From the above studies both for the office building in Breda and the office building in Hoofddorp, we can see that the XGBoost is a more superior model in terms of training data size requirement and accuracy. Therefore, XGBoost is chosen as the desirable model for ΔT and CCV prediction in the final FDD tool.

A comparison of CCV position prediction of building in Hoofddorp as shown in Figure 11 and CCV position prediction of the building in Breda as shown in Figure 15 shows a slight difference in predictions when the actual value is 100%. The building in Hoofddorp uses a changeover coil, whereas the building in Breda uses a separate cooling coil. The presence of a changeover coil can lead to instances where there is transient behaviour (between cooling and heating), and this can lead to incorrect predictions (the predicted value is much lower than the actual value of 100%), as observed in Table 7.

Table 7: ML models performance summary

Building	Variable	Model	R^2 score	RMSE
Office - Hoofddorp	CCV position	XGBoost	0.89	11.5%
		ANN	0.86	12.5%
	ΔT	XGBoost	0.88	0.66K
		ANN	0.88	0.65K
Office - Breda	CCV position	XGBoost	0.85	8.7%
		ANN	0.84	11%
	ΔT	XGBoost	0.93	0.4K
		ANN	0.47	1.15K

6 DBN DEVELOPMENT

For the diagnosis module of the FDD tool, a Diagnostic Bayesian Network (DBN) is developed based on the 4 Symptoms 3 Faults (4S3F) method. The 4 symptoms include energy balance symptoms, energy performance symptoms, operation state symptoms and additional information symptoms. The 3 faults include control faults, model faults and component faults. The DBN is developed only for operational state symptoms, additional information symptoms, operational state faults and component faults for the boundary region around the cooling coil.

For the initial testing of the DBN, the ASHRAE RP-1312 database (Wen, 2011) was used since it has a sufficient and reliable amount of labelled faulty and non-faulty data, which is well-tested and used by many researchers around the world. This initial testing would help to determine if the developed DBN model is reliable and whether it can be used for further development of the FDD tool. The ASHRAE RP-1312 experiments included two identical AHU's which conditioned identical zones in a building, where one AHU is faulty, whereas the other AHU is normal. The AHU is designed in an American configuration which includes an economiser which recirculates the return air as shown in Figure 21.

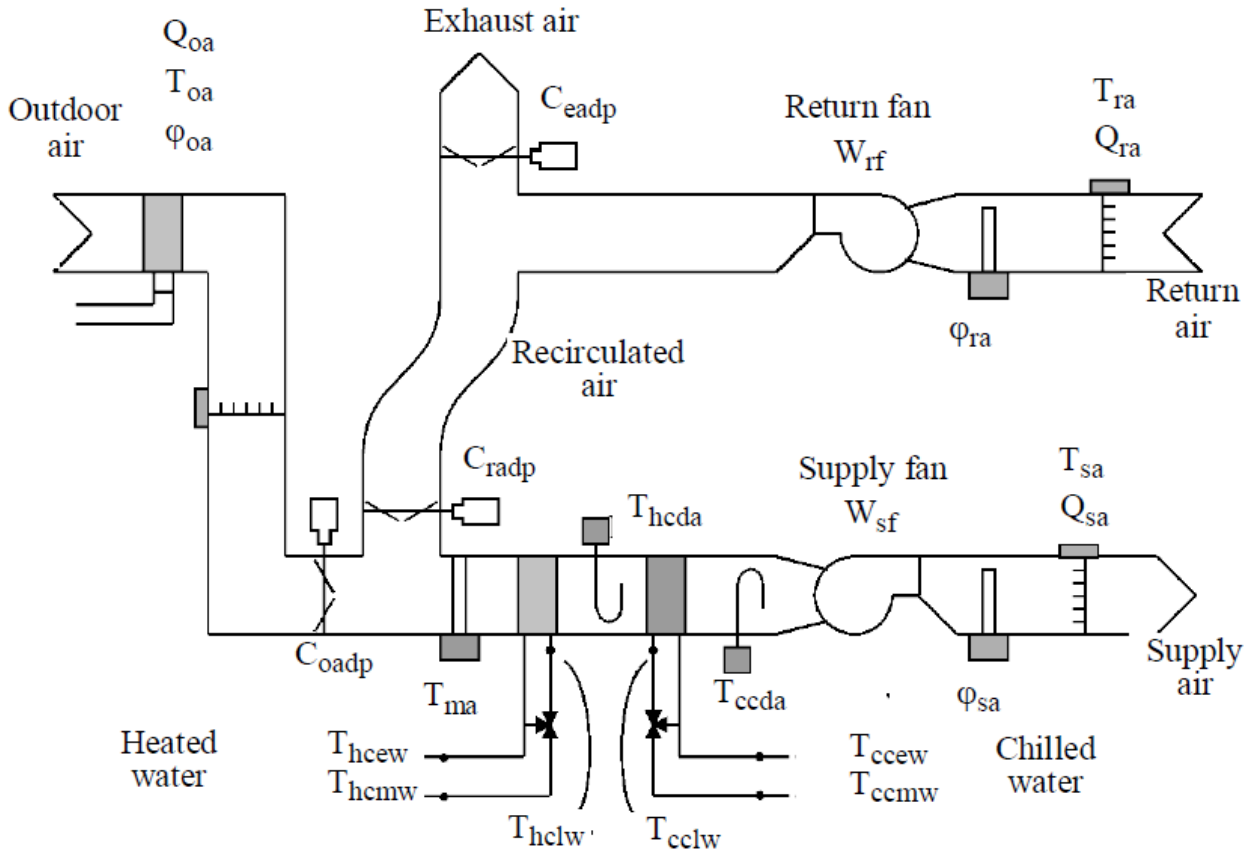

Figure 21: Schematic of AHU used in ASHRAE RP-1312 experiments

The DBN is based on the work of (Zhao et al., 2017) with some minor adjustments made to the network. The prototype of the DBN model was built in GeNie by another PDEng trainee (Karthik Gunderi). GeNie is a statistical model with a user-friendly graphic user interface that allows for interactive model building and learning. It was essential to check the feasibility of the DBN by constructing it in GeNie and then moving to development in python. The main advantage of GeNie is the ability to construct probability distribution tables using Noise-MAX nodes. This reduces the complexity of generating probability tables since it reduces the number of parameters from exponential to linear to the number of parent nodes ($2^{n+1} \rightarrow 2*(n+1)$) (Zhao et al., 2017).

The python framework for the DBN is developed using the python library called pomegranate (Schreiber, 2018) developed at Stanford University. The DBN model, which was developed in GeNie was replicated in python to facilitate easy data transfer between the different modules of the FDD tool. Figure 22 shows the DBN structure developed in python using the pomegranate library. The blue eclipses highlight the fault nodes, whereas the

red eclipses highlight the symptom node. The prior probability distributions were obtained from the work of (Zhao et al., 2015), and the combined probability tables (CPTs) were created based on experience and a general idea of HVAC control loops. Since pomegranate does not have a function to include Noisy-MAX nodes but only normal CPTs, the CPTs were generated from GeNie while inserting probability values for Noisy-MAX nodes and then imported to python.

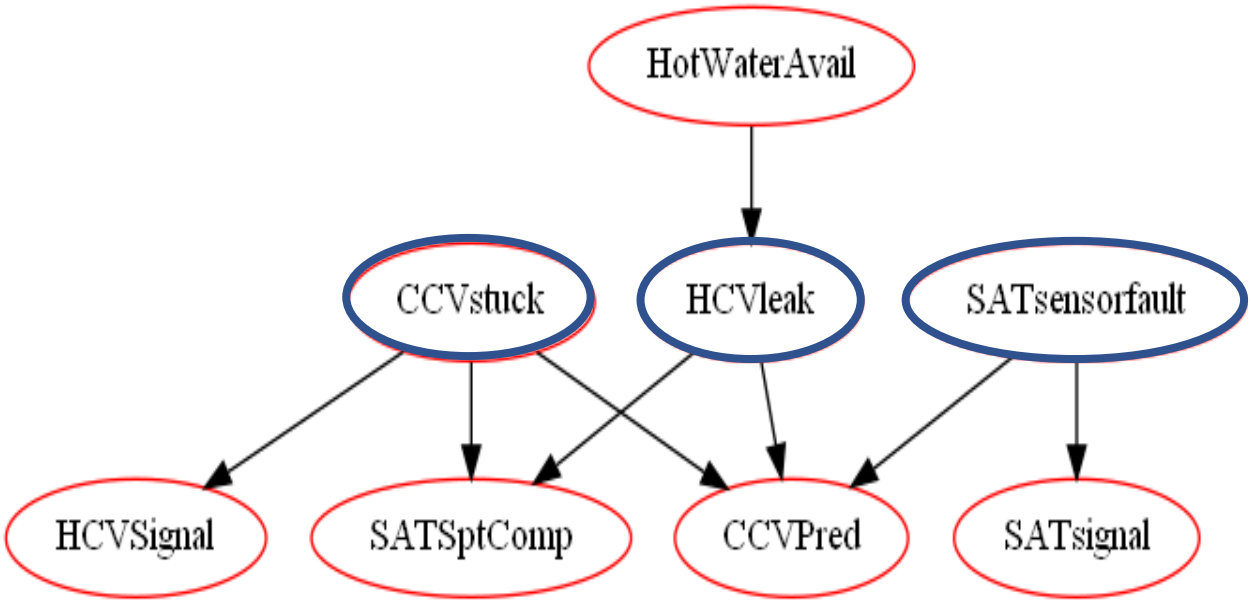

Figure 22: DBN for ASHRAE RP-1312 generated by pomegranate library in python. Blue eclipses highlight fault nodes, red eclipses highlight symptom nodes.

The developed DBN model in python was then used to integrate a fault detection model and eventually display the results of fault detection and diagnosis in a dashboard.

7 CONCLUSIONS AND RECOMMENDATIONS

An early prototype structure of the proposed FDD tool and its design architecture is presented. A business layer of the tool combines state-of-the-art techniques from the AI domain and automates the FDD process. The designed tool is scalable, reliable, rapidly deployable, and interoperable. DBN structure and modelling processes are scaled for both the studied cases possessing different HVAC characteristics. Hence, alluding to generalizability. For further development of the tool following areas have been identified:

- The presented tool does not feature any framework for uniformly identifying building metadata, such as Project Haystack or Brick Schema, which is highly desirable for addressing large-scale deployments.
- Currently, the DBN model doesn't exhibit any learning character, which can be improved by updating conditional probabilities dynamically.

ANNEX 1. PUBLICATIONS AND DISSEMINATION ACTIVITIES

Project deliverables

No	Title
D1.8a	First overview Machine learning software module (beta version)

Publications

Authors	Title	Publisher
Shobhit Chitkara, Alet van den Brink, Shalika Walker, Wim Zeiler	An early prototype for fault detection and diagnosis of Air-Handling Units	Proceedings CLIMA2022, TU Delft, May 23-25
Anand Thamban, Alet van den Brink, Shalika Walker, Wim Zeiler	Detection of the low ΔT syndrome using machine learning models	Proceedings CLIMA2022, TU Delft, May 23-25

Presentations

Date	Title (Presentor)	Event
2022/03/14	TVVL webinar series #4 Predictive and condition-based maintenance (Wim Zeiler |TU/e, Rick Kramer | TUe), Mike van der Heijden | Strukton)	TVVL webinar series THE (Big) Data potential
2021/11/12	TVVL webinar series #2 Fout Detectie & Diagnose (o.a. Wim Zeiler | TU/e) & Dave Baas | Renor)	TVVL webinar series THE (Big) Data potential

ANNEX 2. REFERENCES

Afram A, Janabi-Sharifi F. , 2014Review of modeling methods for HVAC systems. Applied Thermal Engineering 2014;67:507–19. https://doi.org/10.1016/j.applthermaleng.2014.03.055.

Ameisen E. , 2020, Building Machine Learning Powered Applications. O'Reilly UK Ltd.; 2020.

Bruton K, Raftery P, O'Donovan P, Aughney N, Keane MM, O'Sullivan DTJ. , 2014a, Development and alpha testing of a cloud based automated fault detection and diagnosis tool for Air Handling Units. Automation in Construction 2014;39:70–83. https://doi.org/10.1016/j.autcon.2013.12.006.

Bruton K, Raftery P, Kennedy B, Keane MM, O'Sullivan DTJ. , 2014b, Review of automated fault detection and diagnostic tools in air handling units. Energy Efficiency 2014;7:335–51. https://doi.org/10.1007/s12053-013-9238-2.

Bruton K, Coakley D, Raftery P, Cusack DO, Keane MM, O'Sullivan DTJ. , 2014c, Comparative analysis of the AHU InFO fault detection and diagnostic expert tool for AHUs with APAR. Energy Efficiency 2014 8:2 2014;8:299–322. https://doi.org/10.1007/S12053-014-9289-Z.

Chakraborty D, Elzarka H., 2018, Advanced machine learning techniques for building performance simulation: a comparative analysis. Https://DoiOrg/101080/1940149320181498538 2018;12:193–207. https://doi.org/10.1080/19401493.2018.1498538.

Chakraborty D, Elzarka H. , 2019,Early detection of faults in HVAC systems using an XGBoost model with a dynamic threshold. Energy Build 2019;185:326–44. https://doi.org/10.1016/j.enbuild.2018.12.032.

Chen J, Zhang L, Li Y, Shi Y, Gao X, Hu Y., 2022, A review of computing-based automated fault detection and diagnosis of heating, ventilation and air conditioning systems. Renewable and Sustainable Energy Reviews 2022;161:112395. https://doi.org/10.1016/J.RSER.2022.112395.

Chen T, Guestrin C., 2016, XGBoost: A Scalable Tree Boosting System. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016;13-17-August-2016:785–94. https://doi.org/10.1145/2939672.2939785.

Choinière D, Milesi-Ferretti N. , 2014Cost Effective Commissioning of Existing and Low Energy Buildings. 2014.

Fan C, Liu X, Xue P, Wang J., 2021, Statistical characterization of semi-supervised neural networks for fault detection and diagnosis of air handling units. Energy and Buildings 2021;234:110733. https://doi.org/10.1016/J.ENBUILD.2021.110733.

Ferretti NM, Choinière D., 2006, Annex 47 Extended Project Summary 2006. https://doi.org/10.6028/NIST.TN.1750.

Friedman H., PietteM. A., 2001, Comparative guide to emerging diagnostic tools for large commercial HVAC systems. Lawrence Berkeley National Laboratory.

Granderson J, Singla R, Mayhorn E, Ehrlich P, Vrabie D, Frank S., 2017, Characterization and Survey of Automated Fault Detection and Diagnostics Tools | Energy Technologies Area 2017. https://eta.lbl.gov/publications/characterization-survey-automated (accessed March 16, 2021).

Granderson J, Lin G, Singla R, Mayhorn E, Ehrlich P, Vrabie D, et al., 2021, Commercial Fault Detection and Diagnostics Tools: What They Offer, How They Differ, and What's Still Needed 2021. https://doi.org/10.20357/B7V88H.

Gunay HB, Shen W, Yang C., 2018, Text-mining building maintenance work orders for component fault frequency. Building Research & Information 2018. https://doi.org/10.1080/09613218.2018.1459004.

Hu M, Chen H, Shen L, Li G, Guo Y, Li H, 2017, A machine learning bayesian network for refrigerant charge faults of variable refrigerant flow air conditioning system. Energy and Buildings 2018;158:668–76. https://doi.org/10.1016/J.ENBUILD.2017.10.012.

Hüllermeier E, Waegeman W. , 2021, Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Machine Learning 2021;110:457–506. https://doi.org/10.1007/S10994-021-05946-3/FIGURES/17.

Huls AJ. , 2016, Systematic energy performance assessment in operating office buildings Identification and assessment of energy performance gaps in Dutch office buildings, using the Pareto analysis and LEAN Energy Analysis. 2016.

Kim W, Katipamula S., 2018, A review of fault detection and diagnostics methods for building systems. Science and Technology for the Built Environment 2018;24:3–21. https://doi.org/10.1080/23744731.2017.1318008.

Leprince J, Miller C, Zeiler W., 2021, Data mining cubes for buildings, a generic framework for multidimensional analytics of building performance data. Energy and Buildings 2021;248:111195. https://doi.org/10.1016/J.ENBUILD.2021.111195.

Lhotka O., Kyselý J., Farda A., 2018, Climate change scenarios of heat waves in Central Europe and their uncertainties. Theoretical and Applied Climatology, 131(3–4), 1043–1054. https://doi.org/10.1007/s00704-016-2031-3

Li Y, O'Neill Z., 2018, A critical review of fault modeling of HVAC systems in buildings. Building Simulation 2018;11:953–75. https://doi.org/10.1007/s12273-018-0458-4.

Li Y, O'Neill Z., 2019, An innovative fault impact analysis framework for enhancing building operations. Energy and Buildings 2019;199:311–31. https://doi.org/10.1016/j.enbuild.2019.07.011.

Lundberg SM, Lee S-I. , 2017, A Unified Approach to Interpreting Model Predictions. In: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, et al., editors. 2017, Advances in Neural Information Processing Systems 30, Curran Associates, Inc.; 2017, p. 4765–74.

Marquart E., Lange R., 2017, Monitor energiebesparing gebouwde omgeving. Rijksdienst Voor Ondernemend Nederland, 31(0), 6–7. www.RVO.nl./onderwerpen/duurzaam-ondernemen/gebouwen/

Miller C, Arjunan P, Kathirgamanathan A, Fu C, Roth J, Park JY, et al., 2020, The ASHRAE Great Energy Predictor III competition: Overview and results. Science and Technology for the Built Environment 2020;26:1427–47. https://doi.org/10.1080/23744731.2020.1795514.

Mills E., 2011, Building commissioning: A golden opportunity for reducing energy costs and greenhouse gas emissions in the United States. Energy Efficiency, 4(2), 145–173. https://doi.org/10.1007/s12053-011-9116-8

Mo H, Sun H, Liu J, Wei S. , 2019, Developing window behavior models for residential buildings using XGBoost algorithm. Energy Build 2019;205:109564.

Montazeri A,. Kargar S.M., 2020, Fault detection and diagnosis in air handling using data-driven methods. J Build Eng 2020;31:101388.

Najafi M, Auslander DM, Bartlett PL, Haves P, Sohn MD., 2012, Application of machine learning in the fault diagnostics of air handling units. Applied Energy 2012;96:347–58. https://doi.org/10.1016/j.apenergy.2012.02.049.

Nehasil O, Dobiášová L, Mazanec V, Široký J. Versatile, 2021, AHU fault detection – Design, field validation and practical application. Energy and Buildings 2021;237:110781. https://doi.org/10.1016/J.ENBUILD.2021.110781.

Pan B. , 2018, Application of XGBoost algorithm in hourly PM2.5 concentration prediction. IOP Conf Ser Earth Environ Sci 2018;113.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. 2011, Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 2011;12:2825–30.

Pradhan O, Wen J, Chen Y, O'Neill Z.,Wu T, 2021, Pradhan O, Wen J, Lu X.,Chu M., Fu Y., O'Neill Z., Wu T, Selcuk Candan K., 2021, Dynamic bayesian network-based fault diagnosis for ASHRAE guideline 36: high performance sequence of operation for HVAC systems, Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation

Prouzeau A, Dharshini MB, Balasubramaniam M, Henry J, Hoang N, Dwyer T., 2018, Visual Analytics for Energy Monitoring in the Context of Building Management. 2018 International Symposium on Big Data Visual and Immersive Analytics, BDVA 2018 2018. https://doi.org/10.1109/BDVA.2018.8534026.

Schein J, Bushby ST, Castro NS, House JM., 2006, A rule-based fault detection method for air handling units. Energy and Buildings 2006;38:1485–92. https://doi.org/10.1016/j.enbuild.2006.04.014.

Schreiber J., 2018, Pomegranate: Fast and flexible probabilistic modeling in python. Journal of Machine Learning Research, 18, 1–6.

Seyedzadeh S., Rahimian F.P., Glesk I., Roper M., 2018, Machine learning for estimation of building energy consumption and performance: a review. Vis Eng 2018;6.

Shi R, Xu X, Li J, Li Y., 2021, Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization. Applied Soft Computing 2021;109:107538. https://doi.org/10.1016/J.ASOC.2021.107538.

Shi Z, O'Brien W. , 2019, Development and implementation of automated fault detection and diagnostics for building systems: A review. Automation in Construction 2019;104:215–29. https://doi.org/10.1016/j.autcon.2019.04.002.

Taal A, Itard L, Zeiler W., 2018, A reference architecture for the integration of automated energy performance fault diagnosis into HVAC systems. Energy and Buildings 2018;179:144–55. https://doi.org/10.1016/j.enbuild.2018.08.031.

Taal A, Itard L., 2020, Fault detection and diagnosis for indoor air quality in DCV systems: Application of 4S3F method and effects of DBN probabilities. Building and Environment 2020;174:106632. https://doi.org/10.1016/j.buildenv.2019.106632.

Taal A, Itard L., 2020, P&ID-based symptom detection for automated energy performance diagnosis in HVAC systems. Automation in Construction 2020;119:103344. https://doi.org/10.1016/j.autcon.2020.103344.

Verbert K, Babuška R, de Schutter B. ,2017, Combining knowledge and historical data for system-level fault diagnosis of HVAC systems. Engineering Applications of Artificial Intelligence 2017;59:260–73. https://doi.org/10.1016/j.engappai.2016.12.021.

Verhelst J, van Ham G, Saelens D, Helsen L. Model selection for continuous commissioning of HVAC-systems in office buildings: A review. Renewable and Sustainable Energy Reviews 2017;76:673–86. https://doi.org/10.1016/j.rser.2017.01.119.

Walden DD, Roedler GJ, Forsberg K, Hamelin RD, Shortell TM, 2015, International Council on Systems Engineering. Systems engineering handbook : a guide for system life cycle processes and activities. 2015.

Walker S, Khan W, Katic K, Maassen W, Zeiler W., 2020, Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings. Energy and Buildings 2020;209:109705. https://doi.org/10.1016/j.enbuild.2019.109705.

Wang H, Chen Y., 2016, A robust fault detection and diagnosis strategy for multiple faults of VAV air handling units. Energy and Buildings 2016;127:442–51. https://doi.org/10.1016/j.enbuild.2016.06.013.

Wang S., Jiang Z.., 2004, Valve fault detection and diagnosis based on CMAC neural networks. Energy Build 2004;36:599–610.

Wen J, Shun L., 2012, RP-1312 – TOOLS FOR EVALUATING FAULT DETECTION AND DIAGNOSTIC METHODS FOR AIR-HANDLING UNITS. Atlanta, G.A.: 2012.

Wen J. , 2010, Development and validation of a dynamic air handling unit model, Part 2 Human Behavior and Low Energy Architecture: Linking Environmental Adaptation, Personal Comfort, and Energy Use in the Built Environment View project Tools for Evaluating Fault Detection and Diagnostic Methods for HVAC Secondary Systems of a Net Zero Building View project Shun Li. 2010.

Wen J., 2010, Development and Validation of a Dynamic Air Handling Unit Model, Part 1 (RP-1312)(OR-10-007) Human Behavior and Low Energy Architecture: Linking Environmental Adaptation, Personal Comfort, and Energy Use in the Built Environment View project Tools for Evaluating Fault Detection and Diagnostic Methods for HVAC Secondary Systems of a Net Zero Building View project. 2010.

XGBoost Parameters, 2021, https://xgboost.readthedocs.io/en/latest/parameter.html (accessed August 25, 2021).

Xiao F, Zhao Y, Wen J, Wang S., 2014 Bayesian network based FDD strategy for variable air volume terminals. Automation in Construction 2014;41:106–18. https://doi.org/10.1016/j.autcon.2013.10.019.

Yan K, Huang J, Shen W, Ji Z., 2020, Unsupervised learning for fault detection and diagnosis of air handling units. Energy and Buildings 2020;210:109689. https://doi.org/10.1016/j.enbuild.2019.109689.

Yao L, Cai M, Chen Y, Shen C, Shi L, Guo Y. , 2019, Prediction of antiepileptic drug treatment outcomes of patients with newly diagnosed epilepsy by machine learning. Epilepsy Behav 2019;96:92–7.

Yu Y, Woradechjumroen D, Yu D., 2014, A review of fault detection and diagnosis methodologies on air-handling units. Energy and Buildings 2014;82:550–62. https://doi.org/10.1016/j.enbuild.2014.06.042.

Zhang L, Frank S, Kim J, Jin X, Leach M. , 2020, A systematic feature extraction and selection framework for data-driven whole-building automated fault detection and diagnostics in commercial buildings. Building and Environment 2020;186:107338. https://doi.org/10.1016/j.buildenv.2020.107338.

Zhang R, Hong T., 2017, Modeling of HVAC operational faults in building performance simulation. Applied Energy 2017;202:178–88. https://doi.org/10.1016/j.apenergy.2017.05.153.

Zhao Y, Wen J, Wang S., 2015 Diagnostic Bayesian networks for diagnosing air handling units faults – Part II: Faults in coils and sensors. Applied Thermal Engineering 2015;90:145–57. https://doi.org/10.1016/J.APPLTHERMALENG.2015.07.001.

Zhao Y, Wen J, Xiao F, Yang X, Wang S., 2017, Diagnostic Bayesian networks for diagnosing air handling units faults – part I: Faults in dampers, fans, filters and sensors. Applied Thermal Engineering 2017;111:1272–86. https://doi.org/10.1016/J.APPLTHERMALENG.2015.09.121.

Zhao Y, Li T, Zhang X, Zhang C. , 2019, Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future. Renewable and Sustainable Energy Reviews 2019;109:85–101. https://doi.org/10.1016/j.rser.2019.04.021.

Zhou Z-H., 2021, Machine Learning. Springer.