# An early prototype for fault detection and diagnosis of Air-Handling Units

*Shobhit* Chitkara [a], *Alet* van den Brink [b], *Shalika* Walker [a], *Wim* Zeiler [a]

[a] Department of the Built Environment, Eindhoven University of Technology, Eindhoven, the Netherlands, s.chitkara@tue.nl, s.s.walker@tue.nl, w.zeiler@tue.nl.

[b] Afdeling O&T, Kropman Installatietechniek B.V, Nijmegen, the Netherlands, alet.van.den.brink@kropman.nl.

**Abstract.** The built environment is responsible for nearly 35% of energy consumption and is undergoing a digital transformation. Up to 30% of energy is consumed inefficiently due to inadequate setup and/or incomplete utilization of available data. An efficient fault detection and diagnosis (FDD) strategy for air handling units is key to addressing this gap. Even though numerous FDD approaches have been published, real-world applications are far more complex and rarely discussed. This paper deals with FDD tool prototyping and integration aspects and discusses its development for air handling units deployed at 2 case-study buildings located in the Netherlands. The design and development of the FDD tool follows a structured 4 step process. Firstly, literature research is utilized to narrow the design space and establish a complete use case for developing the FDD tool. Secondly, the developed use case is handled utilizing a data-driven strategy to generate fault symptoms using a state-of-the-art extreme gradient boosting algorithm (XGBoost). Thirdly, the detected faults are isolated with a diagnostic Bayesian network. This way the fault detection and diagnosis aspects are separately handled. Lastly, integration of the prototyped tool with a commercially operated continuous monitoring system, currently being utilized to monitor 400 buildings, is discussed. Upon experimental validation, diagnosis specificity exceeding 90% is realized. It is further observed that the prototyped FDD tool could prevent up to 33% of chiller consumed energy. Moreover, the results presented will contribute to the adoption and deployment of AI-based FDD strategies in commercial applications.

**Keywords.** Air Handling Unit, Fault Detection and Diagnosis, Symptoms, Diagnostic Bayesian network, Artificial Intelligence
**DOI**: https://doi.org/10.34641/clima.2022.318

## 1. Introduction

The direct and indirect $CO_2$ emissions from energy use in buildings surpassed 10GtCO2 in 2019, the highest recorded level [1]. Space cooling and heating applications are key drivers of this demand [1]. Despite the efficiency gains due to digitization realized in other sectors such as banking, media etc. this potential is underleveraged in buildings. Up to 30% energy could be saved through the effective use of data collected through deployed continuous monitoring systems (CMS) [2].

Air-Handling Units (AHU) are the most widely studied sub-systems in the Heating, Ventilation, and Air-Conditioning (HVAC) system [2]. In their recent review of Automated Fault Detection and Diagnosis studies (AFDD), Shi et al. in [3] highlighted that there remain limited real-life applications despite much research. Further, the users that have currently adopted state-of-the-art FDD practices typify early adopters or innovators on the technology adoption curve [4].

Considering these aspects, the direct contribution of this paper is to add to the limited real-life demonstrations of FDD and inspire its widespread adoption. More specifically, the design, development, and validation of an FDD tool prototype that utilizes artificial intelligence (AI) methods is discussed. Further, the tool is integrated with a commercially operated CMS that has been utilized to monitor over 400 buildings in the Netherlands.

## 2. Related Work

Granderson et al. in [5] surveyed commercially deployed and under development FDD tools. It can be observed from their survey that FDD tools being utilized by the industry typically rely on a combination of expert rules or first principles. For example, in [6] proposed a cloud-based AFDD tool for AHUs. Their tool utilizes AHU performance assessment rules (APAR) [7]. Some of the common issues identified with using this approach are listed below:

a) Rules-based systems heavily rely on the sensed information. Due to the sensitivity of building owners to initial project costs, most building installations only have sensors limited to their control functionality [8]. Due to the lack of this additional information and the uncertainty associated with deployed sensors, it is difficult to develop reliable quantitative or qualitative models for FDD.

b) The limits utilized for generating alarms using the rule-based approaches are typically set at a higher threshold than desired to minimize the number of false positives [2]. This reduces the ability of an FDD system to detect faults with lower severity.

c) There is a lack of a unified framework for developing generic key-performance indicators (KPIs) and associated detection rules [9].

Bayesian inference-based approaches have been successfully demonstrated by authors of [10] and [11] for HVAC applications. In general, these models can handle circumstances when incomplete, uncertain, or conflicting information is presented as their outputs are fault probabilities instead of Boolean fault outcomes [11].

Besides, for the deployment of advanced algorithms from the machine learning (ML) domain lack of labelled faulty data is a key impediment [12]. Regression model-based or residual generation approaches offer an alternative to working with labelled data [2].

Often, the published research methods utilizing novel FDD techniques start with utilizing a prepared dataset. However, the practical application of these methods with operational CMS is rarely discussed. Granderson et al. [13], in their survey of 14 commercially deployed tools noted that whilst their software stack was proprietary several vendors offer application programming interface (API) to support integration.

Some desirable characteristics proposed by authors of [3] and [14] for an FDD system are categorized as functional and realization design requirements [15], (see **Tab. 1**). They serve as guiding indicators for the prototyped FDD tool.

## 3. Design Methodology

The design methodology for the tool is based on the systems thinking approach. For its software level deployment, Python is utilized as it is now the most popular programming language and has a large collection of continuously maintained packages supporting AI-based development. Further, the proposed software architecture is modular and can be expanded in-depth and at scale. This addresses the scalability and interoperability aspects stated in **Tab. 1**.

**Tab. 1 -** Desirable characteristics for an FDD system

| Functional Aspects | Realization Aspects |
|---|---|
| High accuracy | Adaptability |
| Quick detection and diagnosis | No need for handcrafted AFDD algorithms |
| Robustness | Low Cost |
| Explanation facility | Interoperability |
| Isolability - ability to distinguish between multiple failures | Low storage and computational requirements |
| Novelty identifiability | Limited modelling requirements |
| Heuristic observations as evidence | Automation level in configuration |
| Multiple fault identifiability | Evaluation and decision support capabilities |

In section 3.1, the overall software architecture for the proposed tool is introduced. In section 3.2, the business layer of the proposed tool is revealed, and in section 3.3, the integration aspects are discussed.

### *3.1 FDD Tool Overview*

The overall architecture of the FDD tool is presented in **Fig. 1**. Herein, the implemented workflow is represented with solid arrows. The architecture comprises several layers namely data acquisition, pre-processing, business, post-processing, and visualization. The data acquisition layer is where all aspects concerning data transactions, protocol, and security are maintained.

The design of the FDD tool hereafter can be envisioned as *user agnostic* and *user specific* as shown in **Fig. 1**. The *user agnostic* development concerns design of data preparation and data mining operations on acquired data. The data preparation is carried out using Python libraries such as *Pandas*, and *NumPy*. For data mining, the business layer that wraps the fault detection and diagnosis approach is dwelled upon in section 3.2.

The *user specific* development is to help users efficiently realize outputs from the FDD business layer and enable human-in-the-loop diagnostics. The outputs are post-processed in an intermediary layer before being parsed through to the data visualization layer. The objective of this layer is to provide limited yet valuable evaluation and decision support capabilities for the user as identified in **Tab. 1**. The data visualization layer is designed using an open-sourced web development framework called *Dash by Plotly*. It wraps underlying layers of HTML, JavaScript, and CSS code blocks in a Pythonic syntax. Using this approach, an end-to-end Python-based tool that's reliable, scalable, customizable, and quickly deployable is realized.

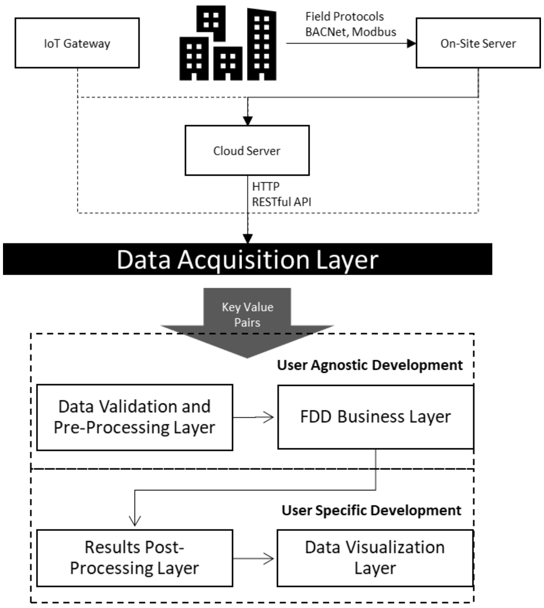


**Fig. 1** - FDD tool prototype architecture

### *3.2 FDD business layer*

For the design of the FDD business layer, 4S3F (four symptoms and three faults) framework is utilized (see **Fig. 2**) [16]. The 4S3F framework combines systems thinking approach with Bayesian probabilistic models. It addresses key aspects such as robustness, isolability, and early fault detection and diagnosis as identified in **Tab. 1.** At its core, the framework is aligned with the design and implementation of the HVAC system, thus making the designed FDD business layer interpretable.

The proposed approach isolates fault diagnosis or root-cause elimination process from fault detection or 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. For example, in this paper, an advanced AI algorithm called XGBoost (extreme gradient boosting) is utilized in the fault detection process [17]. The detected anomalies using XGBoost are then isolated in the diagnosis process.

In the 4S3F framework, the relationship between the faults and the symptoms are characterized using a belief network or also alternatively referred to as the Diagnostic Bayesian network (DBN) [18]. Here the operational state (OS) symptom nodes represent a deviation in the operational state from its expected state. The OS symptoms can be derived from building management system (BMS) data and are further classified as control-based OS indicators and design-based OS indicators [9]. For the demonstrated prototype both control-based and design-based OS indicators have been utilized. The other three kinds of symptom nodes namely Energy Performance (EP), Energy Balance (EB), and Additional (Add.) symptom nodes have been left out of the scope of this paper.

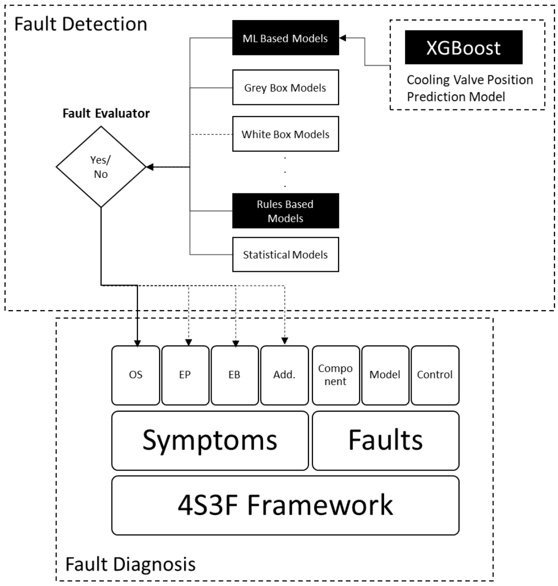


**Fig. 2** - FDD Business Layer

DBN is a directed acyclic graph that encapsulates causal relationships between faults and symptoms in its structure. The DBN structure is further explained by initial beliefs mapped as prior and conditional probability tables [17]. For the faults studied, the prior probabilities are derived using literature research [10]. The conditional probabilities are derived using HVAC expert knowledge. Once, a DBN is finalized the posterior fault probabilities are inferred using the Bayes theorem. Using a sensitivity analysis authors of [18] revealed that if the probability values are reasonably set, the likely diagnosis is not affected by their magnitude.

Faults to be included for developing the initial DBN structure can be prioritized considering the impact of the fault. For AHUs, Gunay et al. in [19] surveyed over 20 years of literature for most studied faults and categorized them by their impact on energy and comfort. From this list, faults namely stuck or leak control valve, fouled or leaking duct, fouled or broken filter, stuck or complete fan failure, and inappropriate supply air setpoints are shortlisted as faults for prototyping the FDD tool. These faults have been prioritized for development considering their high energy impact.

The fault detection layer comprises a modelling layer and a fault evaluation layer. A fault is an unpermitted deviation of at least one characteristic property or feature of the system from an acceptable, usual, standard condition [20]. To realize these acceptable, usual, or standard conditions fault detection models are prepared. These fault detection models can be classified into data-driven and knowledge-driven based methods [9]. The knowledge driven-based methods in this classification are non-AI based methods. Amongst, the data-driven based approaches regression-based approach has been utilized due to the unavailability of fault labels apriori.

For regression modelling, XGBoost is utilized given its superior performance [21]. To select relevant features for the XGBoost algorithm an iterative process is utilized beginning with a coarse feature selection process. This is followed by a wrapper-based feature selection method known as recursive feature elimination and cross-validation (RFECV). Lastly, Shapley additive explanations (SHAP) framework is invoked at the end to further drop features [22]. Minimal possible features are selected to prevent the uncertainty in sensor measurements from spilling over to the inference process.

The FDD business layer is validated by artificially introducing faults in the case study buildings discussed in section 4. For validation of the prototyped FDD tool refer to section 5.2.

### *3.3 FDD Tool Integration*

Besides the veracity of the business logic, a robust coupling between the FDD tool and the deployed CMS is key to its usefulness. The data is acquired over an API, enabled at the server end by the project partner (see **Fig. 1**). For interfacing with remote servers, Python's *requests* package is utilized and is implemented in the data acquisition layer. Here, data is acquired over secure HTTPS. The data acquisition layer has been customized to the case-study buildings, however, it can be expanded to interface directly with on-premises servers or Internet of Things (IoT) gateways. This is the first step toward ensuring the interoperability of the designed system (see **Tab. 1**).

As the proposed tool utilizes an AI-based approach, the software architecture needs to attune to this atypical programming environment. Emmanuel Ameisen in [23] discussed how the development of ML applications at their core comprises two pipelines namely training and inference. For instance, for deploying the XGBoost fault model two data pipelines are stitched. The training pipeline starts with data acquisition over the API. Downstream, includes a key step of filtering out data for AHU's cooling mode operation. Besides, the model is trained until a satisfactory performance is achieved. Cross-validation root mean squared error (CV-RMSE) is utilized as KPI to determine the model's accuracy [24].

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 data pipeline unfolds into two data streams. One is utilized for plotting results from the trained model for visual diagnosis and the other passes straight through to the fault evaluator (ref. **Fig. 2**) and further into the diagnosis pipeline. The training and inference pipelines for other ML models as well as the Bayesian network are designed using a similar approach. To encode the proposed DBN in Python, *pomegranate* package is utilized [25].

## 4. Case Study Description

For validating the prototyped FDD tool two case study buildings located in the Netherlands are utilized. In section 4.1, the first case study, a medium-sized office type building located in the city of Breda is discussed. In section 4.2, the second case study, an educational building located in Nijmegen is discussed.

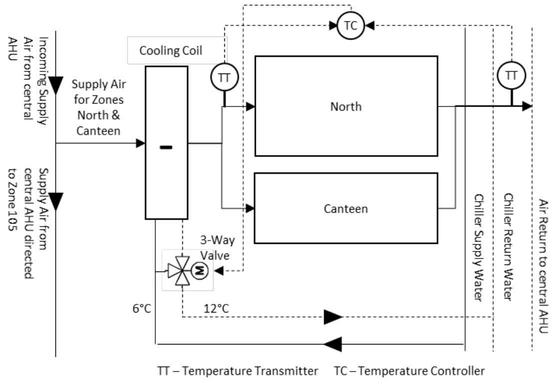


**Fig. 3 -** P&ID Breda office building - North and Canteen Zone

### *4.1 Case study 1: Breda office building*

The office building was commissioned in 1993 and renovated in 2009. The heating and cooling demand for the building is fulfilled by an onsite gas boiler and electric chiller unit respectively. The central AHU supplies three centrally conditioned zones. This constant air volume (CAV) AHU contains a supply and return fan, heating coil, supply, and return filter, and heat recovery wheel (HRW).

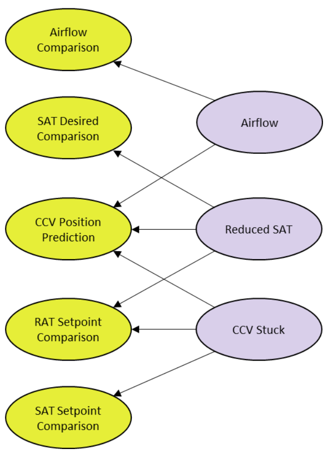


**Fig. 4 -** DBN Breda office building

In the air path of the AHU post supply fan, three individual cooling coils have been placed for each of the supply zones. A portion of the P&I diagram for zones North & Canteen is shown in **Fig. 3**. For this paper, this zone and cooling operation of the AHU is considered. The design of the DBN contained in the FDD business layer emanates from this P&ID subsection. The component fault nodes are depicted in purple in the DBN shown in **Fig. 4** and are in line with the prioritized faults discussed in section 3.2. The airflow fault node is an abstraction for any upstream air side faults in components such as ducts, fans, or filters that can alter the supplied airflow to the zone. Reduced supply air temperature (SAT) and cooling coil valve (CCV) stuck nodes depict reduced setpoint and cooling coil control valve stuck faults respectively.

**Tab. 2 –** DBN symptom nodes and states description

| # | Symptom node | Symptom state | Rules for setting the state |
|---|---|---|---|
| 1 | Airflow Comparison | High | $F_{act}-F_{pred}>\Theta$ |
| | | Low | $F_{act}-F_{pred}<-\Theta$ |
| | | Fault-free | $F_{act}-F_{pred}<=\Theta$ |
| 2 | SAT Desired Comparison | Negative | $T_{set}-T_{set,des}<-\Theta$ |
| | | Fault-free | $T_{set}-T_{set,des}<=\Theta$ |
| 3 | CCV Prediction | Positive | $X_{ccv}-X_{ccv,pred}>\Theta$ |
| | | Negative | $X_{ccv}-X_{ccv,pred}<-\Theta$ |
| | | Fault-free | $X_{ccv}-X_{ccv,pred}<=\Theta$ |
| 4 | RAT Setpoint Comparison | Positive | $T_{ra}-T_{ra,set}>\Theta$ |
| | | Negative | $T_{ra}-T_{ra,set}<-\Theta$ |
| | | Fault-free | $T_{ra}-T_{ra,set}<=\Theta$ |
| 5 | SAT Setpoint Comparison | Positive | $T_{sa}-T_{sa,set}>\Theta$ |
| | | Negative | $T_{sa}-T_{sa,set}<-\Theta$ |
| | | Fault-free | $T_{sa}-T_{sa,set}<=\Theta$ |

**Key:**

**F** - Flow Rate in m3/s, **T** - Temperature in °C, **X** - Control Position in %, **Θ** – Threshold, **act** – Actual, **pred** – Predicted, **ccv** – Cooling coil valve, **des** – Desired, **sa** – Supply air, **ra** – Return Air, **set** – Setpoint

The symptom nodes depicted in yellow in **Fig. 4** are generated using a combination of multiple modelling approaches. The rules for passing evidence to the symptom nodes are enlisted in **Tab. 2**. Airflow comparison node and CCV position prediction nodes are activated using predictions from a statistical and ML model respectively. Features selected for training the ML model are provided in **Tab. 3**.

### *4.2 Case study 2: Nijmegen school*

The second building is a school located in Nijmegen. It was commissioned in the year 2010. The HVAC installation at the building comprises an Aquifer Thermal Energy Storage (ATES) system supported by a heat pump on the generation side. On the distribution side, two AHUs are installed. The AHUs operate with a CAV control strategy. Their supply air temperature is maintained using two-way control valves that throttle supply water through a common cooling and heating coil. In comparison with the discussed office building, here a single casing houses all of the AHU components.

**Tab. 3 -** XGBoost model features

| Breda office building | Nijmegen school |
|---|---|
| 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 | Supply water temperature (coil), Suction air temperature, Return water temperature (coil), Supply air temperature (SAT) post-HRW, SAT setpoint, Supply & Return pressure drop, Supply airflow speed, Week of year, SAT post supply fan, Return air temperature, SAT post coil, Control HRW, HRW (on/off) |

As with case study 1, the operation of the AHU in its cooling mode operation is considered. The DBN for the cooling operation emanates from the P&ID of the installed HVAC system. Despite the differences in HVAC configuration between the two case studies, the DBN structure does not differ beyond the Reduced SAT fault node shown in **Fig. 4**. This alteration is required since the supply air temperature setpoint is feed-forward controlled based on the outdoor air temperature alone. In contrast at the Breda office, the zone supply air temperature setpoint is controlled using two variables return air and outdoor air temperatures. From the symptom nodes listed in **Tab. 2**, the RAT setpoint comparison node is excluded, and the rest are retained in the developed DBN for this case.

## 5. Designed FDD tool and its experimental validation

In this section, an early prototype of the FDD tool is presented. In section 5.1, the dashboard layout is shown. Thereafter in section 5.2, results from experimental validation of the underlying FDD business layer are discussed. Lastly in section 5.3, a savings analysis to estimate the benefit of deploying the FDD tool is presented.

### *5.1 FDD Tool*

The developed FDD application utilizes the React.js framework in its underlying layers and is accessed using a standard web browser interface.

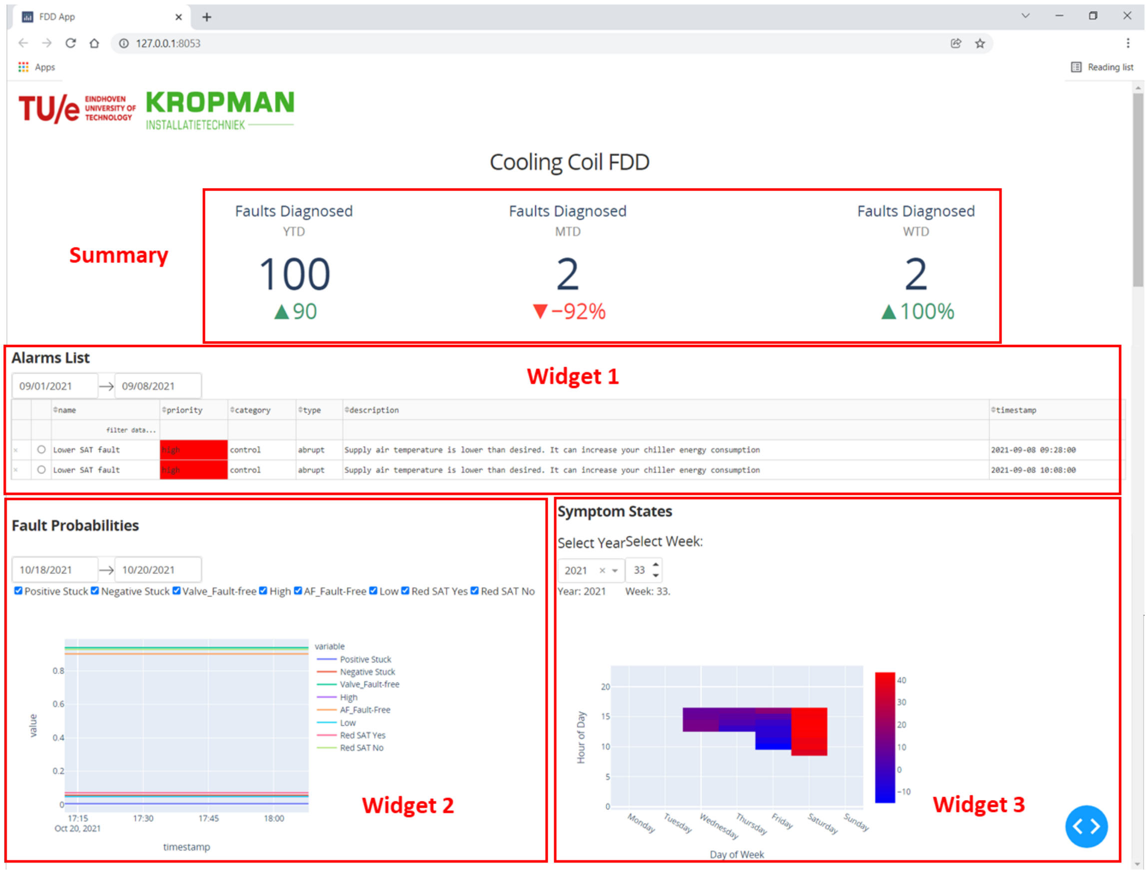


**Fig. 5 -** Designed FDD Dashboard - Case study one

The application is currently deployed locally, however, is extensible to a cloud environment. A dashboard for the cooling coil diagnosis at Breda with its key performance indicators are presented across multiple widgets in **Fig. 5**. The data presented in the figure is for representation purposes alone. In the *summary widget*, summary statistics describing the faults diagnosed by the FDD tool across various time horizons such as year to date (YTD) are shown. In *widget 1*, faults diagnosed by the FDD tool are presented as alarms for users. Alongside each alarm, qualitative information such as fault priority, category, type, description about the raised alarm is provided. Further, the user is equipped with filtering, searching, and sorting functionalities for an enhanced user experience. In *widget 2*, the probabilities of various fault states are plotted as time series. In *widget 3* using a heat map representation, the intensity of the residual generated using the XGBoost model can be found. *Widgets 2 & 3* have been developed to augment the capabilities of a building manager to quickly identify anomalies and aid human-in-the-loop diagnostics.

### *5.2 FDD Validation Results*

For validating the discussed FDD business layer (see section 3.2), experiments were carried out at both buildings. During these experiments, faults were artificially introduced into the system to simulate faulty behaviour. In this paper results from introducing two faults (one in each case study) are discussed.

On 16th July 2021, an experiment was carried out and a stuck valve fault was introduced in the north zone cooling coil at Breda. The position of the three-way control valve that regulates the flow was fixed at 75% at 15:30 using a BMS override. It can be observed from **Fig. 6** that the predicted valve position deviates significantly from the measured actual valve position post introduction of this fault and generates a large residual. Also, the computed probability of the fault state (Positive Stuck) changes from a low (less than 5%) to a high likelihood (~70%) of fault presence. The user can observe this probability shift and change in residual using *widget 2* and *widget 3* respectively presented in **Fig. 5**. An alarm generated automatedly under such an event can be found in *widget 1* in **Fig. 5**.

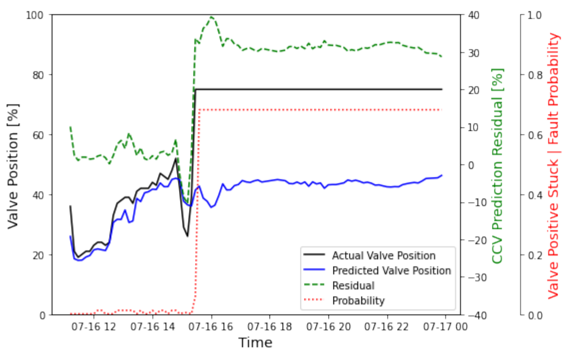


**Fig. 6 -** Stuck valve experiment at Breda

On 3rd August 2020, the supply air setpoint was reduced to 17°C to simulate a reduced setpoint fault scenario at AHU one at the Nijmegen school. As can be observed from **Fig. 7** the desired setpoint given the prevailing outdoor air temperature should have neared 21°C under a normal scenario. However, the automated setpoint determining logic was manually overwritten in the BMS to introduce this fault. The designed DBN was able to confirm (with more than 85% likelihood) this fault when a large positive residual is observed on the valve position prediction state.

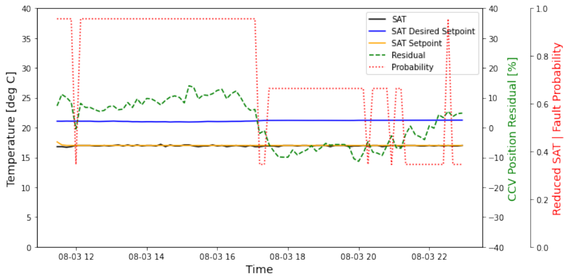


**Fig. 7 -** SAT setpoint experiment at Nijmegen

Besides the reported experiments additional experiments spread over nearly two months were carried out to validate the prototyped tool across multiple faults scenarios at both locations. During this period, diagnosis sensitivity of 67% and specificity of 92% were recorded at the Breda office. At Nijmegen school diagnosis sensitivity of 84% and specificity of 94% were observed. The realized results validate the veracity of the developed FDD business layer for both locations.

### *5.3 Potential Savings: Breda office building*

Through the validation process faults introduced in the system were successfully diagnosed with the tool. Using the case presented in **Fig. 6**, preventable energy waste is estimated. The fault was introduced late Friday afternoon and corrected the following Monday. The chiller's energy consumption between the periods 16:00-17:00 was compared on both days to estimate the energy savings. Using an energy meter, a 63% increase in the chiller's energy consumption was measured during the faulty operation. As faults were simultaneously introduced in all cooling coil control valves, the measured increase is apportioned in the ratio of airflow through each zone. Using this approach, nearly a 33% increase in the chiller's energy consumption is estimated as attributable to the stuck valve fault in the north zone's cooling coil control valve.

## 6. Conclusion

An early prototype of the proposed FDD tool along with its design architecture is presented. Further, integration with an operational CMS is demonstrated for two case study buildings. The 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.

Upon experimentally validating the designed tool, encouraging results (diagnosis specificity exceeding 90%) across two use-case environments are obtained. The continuous monitoring setup at these installations can be augmented with the prototyped tool and energy penalties due to faults can be avoided. At the cooling coil installation at Breda, nearly 33% of energy savings are estimated, and consequent additional emissions are thus preventable.

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.

## 7. Acknowledgement

The research work is funded by the TKI Urban Energy project 'Continuous commissioning of lowdT'; and the Eindhoven Engine project 'CM-HVACFDD'. The work is also supported by Kropman Installatietechniek. Special thanks to Joris de Ruiter, Jan-Willem Dubbeldam, John Verlaan, and Roel Vos at Kropman for facilitating the experiments, data collection and software integration process.

## 8. References

[1] IEA. Tracking Buildings 2020. Paris; 2020.

[2] Kim W, Katipamula S. A review of fault detection and diagnostics methods for building systems. Science and Technology for the Built Environment. 2018 Jan 2;24(1):3–21.

[3] Shi Z, O'Brien W. Development and implementation of automated fault detection and diagnostics for building systems: A review. Vol. 104, Automation in Construction. Elsevier B.V.; 2019. p. 215–29.

[4] Lin G, Kramer H, Granderson J. Building fault detection and diagnostics: Achieved savings, and methods to evaluate algorithm performance. Building and Environment. 2020 Jan 15;168:106505.

[5] Granderson J, Singla R, Mayhorn E, Ehrlich P, Vrabie D, Frank S. Characterization and Survey of Automated Fault Detection and Diagnostics Tools | Energy Technologies Area [Internet]. 2017 [cited 2021 Mar 16]. Available from: https://eta.lbl.gov/publications/characterization-survey-automated

[6] Bruton K, Raftery P, O'Donovan P, Aughney N, Keane MM, O'Sullivan DTJ. Development and

alpha testing of a cloud-based automated fault detection and diagnosis tool for Air Handling Units. Automation in Construction. 2014 Apr 1;39:70–83.

[7] Schein J, Bushby ST, Castro NS, House JM. A rule-based fault detection method for air handling units. Energy and Buildings. 2006 Dec 1;38(12):1485–92.

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

[9] Taal A, Itard L. P&ID-based symptom detection for automated energy performance diagnosis in HVAC systems. Automation in Construction. 2020 Nov 1;119:103344.

[10] Zhao Y, Wen J, Wang S. Diagnostic Bayesian networks for diagnosing air handling units faults – Part II: Faults in coils and sensors. Applied Thermal Engineering. 2015 Nov 5;90:145–57.

[11] Xiao F, Zhao Y, Wen J, Wang S. Bayesian network based FDD strategy for variable air volume terminals. Automation in Construction. 2014 May 1;41:106–18.

[12] Fan C, Liu X, Xue P, Wang J. Statistical characterization of semi-supervised neural networks for fault detection and diagnosis of air handling units. Energy and Buildings. 2021 Mar 1;234:110733.

[13] Granderson J, Lin G, Singla R, Mayhorn E, Ehrlich P, Vrabie D, et al. Commercial Fault Detection and Diagnostics Tools: What They Offer, How They Differ, and What's Still Needed. 2021 Jun 28 [cited 2021 Oct 19]; Available from: https://escholarship.org/uc/item/4j72k57p

[14] Yu Y, Woradechjumroen D, Yu D. A review of fault detection and diagnosis methodologies on air-handling units. Vol. 82, Energy and Buildings. Elsevier Ltd; 2014. p. 550–62.

[15] Zeiler W. Basisboek Ontwerpen. Noordhoff Publishers; 2014.

[16] Taal A, Itard L, Zeiler W. A reference architecture for the integration of automated energy performance fault diagnosis into HVAC systems. Energy and Buildings. 2018 Nov 15;179:144–55.

[17] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining [Internet]. 2016 Mar 9 [cited 2021 Aug 19];13-17-August-2016:785–94. Available from: https://arxiv.org/abs/1603.02754v3

[18] Hu M, Chen H, Shen L, Li G, Guo Y, Li H, et al. A machine learning bayesian network for refrigerant charge faults of variable refrigerant flow air conditioning system. Energy and Buildings. 2018 Jan 1;158:668–76.

[19] Taal A, Itard L. Fault detection and diagnosis for indoor air quality in DCV systems: Application of 4S3F method and effects of DBN probabilities. Building and Environment. 2020 May 1;174:106632.

[20] Gunay B, Shen W, Yang C. Characterization of a Building's operation using automation data: A review and case study. Vol. 118, Building and Environment. Elsevier Ltd; 2017. p. 196–210.

[21] Isermann R. Fault-diagnosis systems: An introduction from fault detection to fault tolerance. Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. Springer Berlin Heidelberg; 2006. 1–475.

[22] Miller C, Arjunan P, Kathirgamanathan A, Fu C, Roth J, Park JY, et al. The ASHRAE Great Energy Predictor III competition: Overview and results. Science and Technology for the Built Environment [Internet]. 2020;26(10):1427–47. Available from: https://doi.org/10.1080/23744731.2020.1795514

[23] Lundberg SM, Lee S-I. A Unified Approach to Interpreting Model Predictions. In: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, et al., editors. Advances in Neural Information Processing Systems 30 [Internet]. Curran Associates, Inc.; 2017 [cited 2021 Nov 26]. p. 4765–74. Available from: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[24] Ameisen E. Building Machine Learning Powered Applications. O'Reilly UK Ltd.; 2020. 250.

[25] Walker S, Khan W, Katic K, Maassen W, Zeiler W. Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings. Energy and Buildings. 2020 Feb 15;209:109705.

[26] Schreiber J. Pomegranate: fast and flexible probabilistic modeling in python. 2017 Oct 31;

### *Data Statement*

The datasets generated during and/or analysed during the current study are not publicly available because they are provided by a company and has to be kept private owing to privacy and security concerns but are/will be available if the concerned parties contact us and explain the reason for the data requirement. However, the final decision lies with the company and access cannot always be guaranteed.