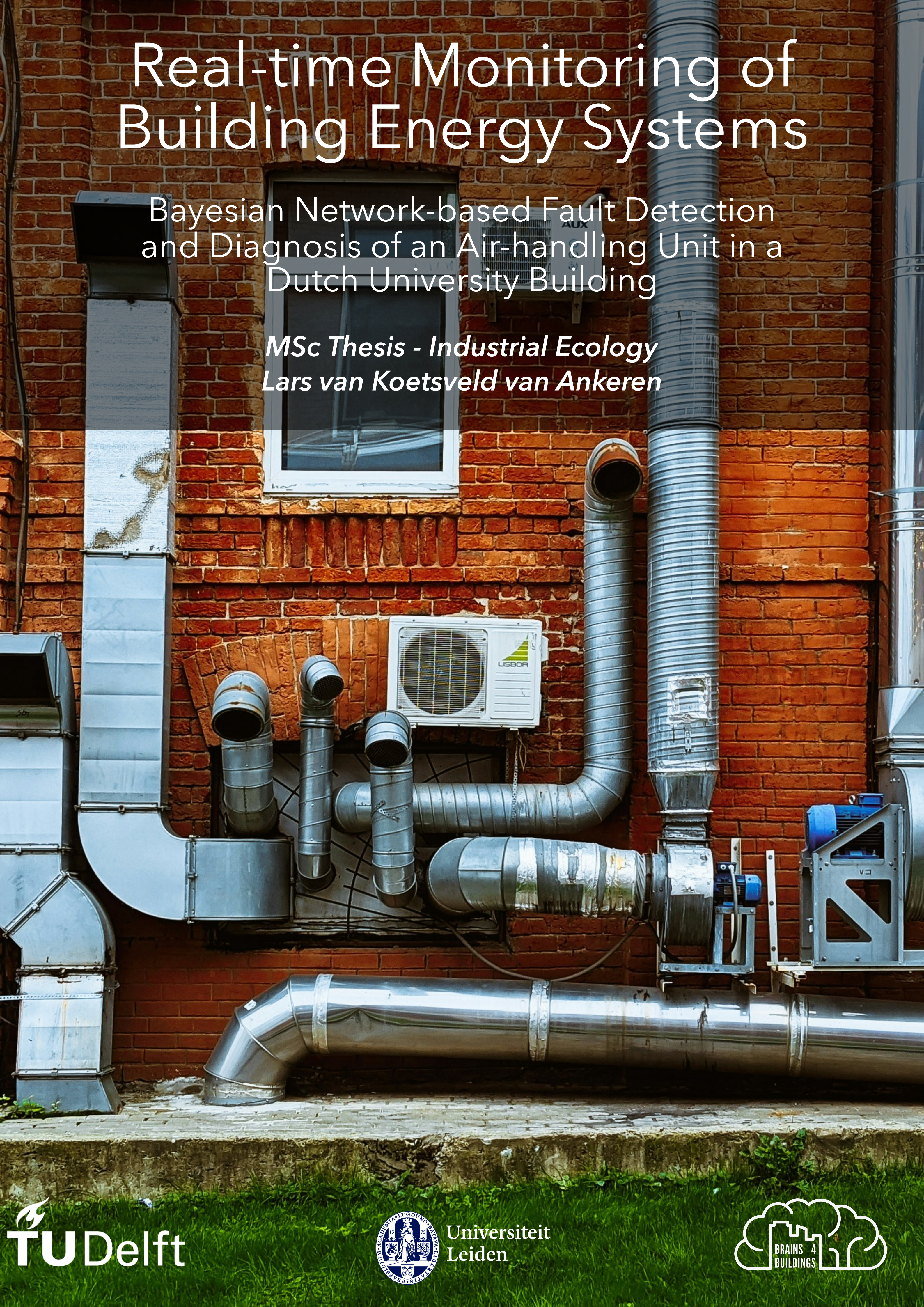
Real-time Monitoring of Building Energy Systems

Bayesian Network-based Fault Detection and Diagnosis of an Air-handling Unit in a Dutch University Building

MSc Thesis - Industrial Ecology
Lars van Koetsveld van Ankeren



TU Delft
Universiteit Leiden
Brains 4 Buildings

Real-time Monitoring of Building Energy Systems

Bayesian Network-based Fault Detection and Diagnosis of an Air-handling Unit in a Dutch University Building

Lars van Koetsveld van Ankeren

Leiden Student ID
3036901

Delft Student ID
4676351

First Supervisor
Laure Itard

Second Supervisor
Neil Yorke-Smith

Daily Supervisor
Chujie Lu

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Industrial Ecology at Delft University of Technology and Leiden University

To be defended publicly on
August 23$_{rd}$ *2024*

Acknowledgements

This thesis has been written with the help of many people, for which I would like to express my gratitude.

First of all, I would like to thank my supervisors for all of the guidance and feedback you have given me. Thank you Laure for being my first supervisor, and introducing me to this topic. I did not expect to do my thesis at the architecture faculty a year ago, but I have certainly enjoyed it. To Neil as well, thank you for guiding me on which parts of my research to focus on. I am glad that you were willing to supervise an Industrial Ecology thesis, somewhat different from what you may have been used to. Finally, thank you Chujie for all of the discussions we have had on how to further improve the model, and all the work you did on setting up the experiments. Working together with you for the past half year has been great.

I would also like to express my thanks to Arie, for your great help in setting up the fault detection model. Additionally, thank you Ziao and Martin, for showing me your diagnostic models and thinking along with me.

Finally, I want to say thank you to all of my friends and family who supported me during the writing of the thesis. And to Iris, thank you for all of your kind support this past year.

Lars van Koetsveld van Ankeren
Delft, August 2024

Summary

A significant part of worldwide energy consumption is used to provide a comfortable climate inside buildings. This energy is mostly used by heating, ventilation and air-conditioning (HVAC) systems. A part of this energy is currently wasted due to faults, such as incorrect control signals or component failures. In the field of HVAC fault detection and diagnosis (FDD), methods are developed that aim to detect that a fault is present in the system, and consequently diagnose which fault this is.

However, most of the methods that are currently being developed are data-driven, which require large amounts of labelled data. In practice, this data often is not available, leading to a lack of adoption of FDD methods by building operators. Additionally, most of the FDD approaches proposed in the literature have not been tested in real-time, instead validation has been performed on already existing datasets.

In this thesis, an FDD method is proposed to diagnose part of the HVAC system, air-handling units (AHUs), in real-time. Air-handling units are an important part of the HVAC system, responsible for a significant part of the energy consumption. The method applies the four symptoms three faults (4S3F) framework, to construct a diagnostic Bayesian network (DBN) without relying on historical data for symptom detection or fault diagnosis.

This DBN is implemented in Python to diagnose a case study AHU in a Dutch university building. The method was validated with fault experiments, which were diagnosed with diagnosis periods ranging from one 10-minute sample to three hours. All of the faults that were introduced and included in the DBN were accurately diagnosed for all diagnosis periods. For the AHU operation data of 2022 and 2023, mostly control and temperature faults were diagnosed. Incorrect control of the heating coil valve was found in 25% and incorrect control of the ERW in 11% of the days considered. Additionally, a deviation between the measured intended supply temperature of more than 1 ° was found for 79% of the days in the dataset.

Finally, the application of the method for real-time fault diagnosis has also been demonstrated, by collecting sensor data through the data streaming platform Apache Kafka. This data was consequently stored locally in the time-series database InfluxDB, which could be queried for performing the diagnosis. The processing time of performing the diagnosis was approximately ten seconds, demonstrating the potential for performing diagnoses per sample.

Recommendations for future work include extending the DBN to diagnose faults related to the cooling coil and faults occurring outside of operation hours. Additionally, the developed diagnosis method should be applied to other AHUs, to further research the generalisability of the DBN.

Contents

Preface **i**

Summary **ii**

Nomenclature **v**

1 Introduction **1**
- 1.1 Problem analysis . . . 1
- 1.2 Research aim . . . 2
- 1.3 Research context . . . 2
- 1.4 Thesis outline . . . 3

2 Literature review and theoretical background **4**
- 2.1 Literature review . . . 4
  - 2.1.1 Fault diagnosis methodologies . . . 4
  - 2.1.2 Diagnostic Bayesian networks . . . 5
  - 2.1.3 Air-handling unit diagnosis . . . 6
  - 2.1.4 4S3F framework . . . 7
- 2.2 Theoretical background . . . 10
  - 2.2.1 Description of air-handling units . . . 10
  - 2.2.2 Bayesian network theory . . . 11
  - 2.2.3 Building management systems . . . 12

3 Descriptive analysis of case study and historical data **13**
- 3.1 Case study description . . . 13
- 3.2 Historical data analysis . . . 16

4 FDD model development **19**
- 4.1 AHU faults . . . 19
- 4.2 DBN description . . . 19
  - 4.2.1 Fault diagnosis . . . 19
  - 4.2.2 Symptom rules . . . 22
  - 4.2.3 Conditional probabilities . . . 26
- 4.3 Data preprocessing . . . 27

5 Experimental validation **30**
- 5.1 Description of experiments . . . 30
- 5.2 Diagnosis setup . . . 33
- 5.3 Diagnosed faults . . . 35
- 5.4 Impact of the diagnosis period . . . 37
- 5.5 Sensitivity analysis . . . 38

6 Diagnosis of historical data **40**
- 6.1 Diagnosis setup . . . 40
- 6.2 Diagnosis results . . . 40
  - 6.2.1 Detected symptoms . . . 41

6.2.2 Diagnosed faults 43
6.3 Sensitivity analysis 45

7 Real-time implementation 49
7.1 Description of Python frameworks 49
7.2 Python implementation 50
7.3 Real-time diagnosis results 53

8 Discussion 57
8.1 Experimental results 57
8.2 Historical results 58
8.3 Real-time results 58
8.4 Generalisability of the diagnosis method 59
8.5 Sustainability implications 60

9 Conclusion 61
9.1 Limitations 62
9.2 Recommendations for future work 62

References 64

A Fault experiments 69

B Revised DBN 70

C Python packages 73

D Example Kafka message 74

Nomenclature

Abbreviations

Abbreviation	Definition
AHU	Air handling unit
BMS	Building management system
CAV	Constant air volume
CC	Cooling coil
CCV	Cooling coil valve
CSV	Comma-separated values
DBN	Diagnostic Bayesian network
EP	Energy performance
ERW	Energy recovery wheel
FDD	Fault detection and diagnosis
HC	Heating coil
HCV	Heating coil valve
HVAC	Heating, ventilation and air-conditioning
OS	Operational state
UTC	Coordinated Universal Time
VAV	Variable air volume

Symbols

Symbol	Definition	Unit
T_o	Outdoor air temperature	°C
T_i	Inlet air temperature	°C
T_p	Preheated air temperature	°C
T_s	Supply air temperature	°C
T_r	Return air temperature	°C
T_e	Exhaust air temperature	°C
p_s	Supply air static pressure	Pa
p_r	Return air static pressure	Pa
Δp_s	Pressure difference at supply filter	Pa
Δp_r	Pressure difference at return filter	Pa
T_{set}	Supply air temperature setpoint	°C
$p_{s,set}$	Supply pressure setpoint	Pa
$p_{r,set}$	return pressure setpoint	Pa
U_{hcv}	Heating coil valve control	%
U_{ccv}	Heating coil valve control	%
U_{erw}	Energy recovery wheel control	%
$U_{s,fan}$	Supply fan control	%
$U_{r,fan}$	Return fan control	%
A_{hc}	Heating coil pump alert	Boolean
A_{erw}	Energy recovery wheel alert	Boolean
$A_{s,fan}$	Supply fan failure alert	Boolean
$A_{r,fan}$	Return fan failure alert	Boolean
$A_{s,filter}$	Supply filter alert	Boolean
$A_{r,filter}$	Return filter alert	Boolean

1
Introduction

This chapter introduces the thesis, by first providing an analysis of the investigated problem. This is followed by the research aim and formulated research questions. Then, the context of this thesis in relation to the Industrial Ecology program and the Brains4Buildings research project is described. The chapter ends with an overview of the remaining chapters in this thesis.

1.1. Problem analysis

In the European Union (EU), buildings are estimated to account for 40% of the total energy consumption [52]. This has to be drastically reduced, as the EU has the objective of reducing greenhouse gas (GHG) emissions by at least 60% in the building sector by 2030 compared to 2015 [14]. Furthermore, the EU aims to have fully decarbonised the entire building stock in 2050 [14].

A large portion of the energy used in buildings is used by heating, ventilation, and air-conditioning (HVAC) systems to provide a comfortable indoor climate. in the USA, these systems account for roughly half of the energy consumed inside buildings [41]. Up to 30% of the energy used by HVAC systems could be wasted due to faults [18]. This makes reducing this energy waste critical for lowering energy usage and consequently avoiding carbon emissions.

The field of HVAC fault detection and diagnosis (FDD) aims to design methods for identifying faults in HVAC systems. One of the components that FDD is often applied to is the air-handling unit (AHU) [7]. These units play an important role within HVAC systems, as they provide airflow as well as heating and cooling. As a result, they consume a significant proportion of the HVAC system's energy [7].

Although much research has been done on different HVAC FDD methods, there is a lack of research on FDD applications in practice. Most research of the last two decades has focused on data-driven methods, with 79% of the published papers between 1998 and 2018 having applied data-driven methods [60]. Data-driven methods require large amounts of labelled data, as well as faulty and fault-free datasets. In practice, this data is often unavailable, leading to a lack of adoption of FDD methods by building operators [29].

An alternative methodology for FDD is the knowledge-based approach, which does not require data for setting up the diagnostics. A recent framework to perform knowledge-based diagnosis has been proposed by Taal et al., referred to as the four symptoms three faults (4S3F) framework [49]. The 4S3F framework applies a diagnostic Bayesian network (DBN) to infer probabilities of faults occurring based on detected symptoms [48]. In this way, symptom detection is separated from fault diagnosis, which is similar to how engineers diagnose faults in HVAC systems [49].

However, there is a lack of applications of this framework in practice. In general, limited real-life applications of methods proposed in the literature exist [29], especially for Europe and the Netherlands [1]. The datasets most frequently used are from China and the United States, which have different climate conditions compared to Europe [35]. Additionally, most of the methods proposed in the literature do not focus on real-time implementation of the diagnosis method [2]. The methods are mostly validated in small-scale HVAC systems or simulated environments [7].

Finally, the majority of research has concentrated on AHUs without energy recovery wheels (ERWs), whereas energy recovery is mandatory for newly built AHUs in Europe [15]. Furthermore, the necessity for research on AHUs with an ERW has increased as a consequence of the controversy surrounding AHUs without ERWs, following the global COVID-19 pandemic [30].

1.2. Research aim

This thesis aims to bridge the gap between theory and practice, by implementing a Bayesian network-based method for real-time fault diagnosis of AHUs. The fault diagnosis method is implemented and validated for a case study AHU in a Dutch university building.

The main research question is therefore formulated as the following:

'How can Bayesian network-based fault diagnosis accurately find faults for air-handling units in real-time based on expert knowledge?'

In order to answer the main research question, it is divided into the following sub-questions:

- 'How can expert knowledge be transferred to a diagnostic Bayesian network to detect and diagnose common air-handling unit faults?'
- 'Which faults are diagnosed with Bayesian network-based fault detection and diagnosis during normal operation of air-handling units?'
- 'How can Bayesian network-based fault detection and diagnosis be applied to diagnose air-handling unit faults in real-time?'

For the first sub-question, the detection of symptoms and diagnosis of faults are researched to find a DBN that can diagnose AHU faults accurately based on expert knowledge. For this symptom detection rules and relationships between symptoms and faults need to be determined.

The second sub-question relates to the faults that can occur during the operation of the AHU. This can give an insight into the frequency of different faults occurring in practice, which have an impact on the energy consumption of the AHU.

The third sub-research question targets the real-time aspect specifically. The use of data streaming solutions for fault diagnosis is researched. Additionally, the impact of fault diagnosis in different time intervals will be investigated.

1.3. Research context

This thesis is carried out as part of Brains4Buildings, a multiple-year research project coordinated by TU Delft. It is a collaboration between 39 partners from 'knowledge institutions, installation companies, energy consultancies, platform/interface developers, building owners and managers, technology suppliers and industry associations' [4]. One of the aims of the project is to develop self-diagnostic systems for improving energy efficiency and smart maintenance of buildings, which this thesis also contributes to. The results of this thesis are shared with partners of the research project, enabling practical application of the results.

A number of studies have already been carried out for Brains4Buildings related to this thesis. The paper published by Chitkara et al. applied 4S3F to an AHU system in the Netherlands [9], based on his EngD thesis [8]. Wang et al. describe DBNs for AHUs in different sensor environments [57]. Anand et al. applied 4S3F to a chiller unit [51], also based on his EngD thesis [50]. Finally, the MSc thesis by Heidtmann extended 4S3F to include occupancy [21], and the MSc thesis of Luo diagnosed temperature sensors with 4S3F [33].

This thesis is submitted for the degree of Master of Science in Industrial Ecology. The Industrial Ecology context of this thesis is mainly related to the aim of reducing energy waste in HVAC systems. Despite the existence of fault diagnosis methods with high accuracy that have been developed in the literature, their lack of adoption in practice has resulted in minimal energy savings. This creates a need for a real-time implementation that takes into account the limitations that occur in practice, such as the absence of sensors and the limited availability of data. The potential energy savings of automated fault diagnosis are significant, which is contributed to by the method developed in this thesis.

Additionally, developing the fault diagnosis method requires a multidisciplinary approach. Technical knowledge of the energy flows within the HVAC system is required, as well as data science knowledge for real-time data processing. Furthermore, fault diagnosis also relies on engineering knowledge of the operation of AHU systems. Therefore, conversations with building management are required for formulating the symptom detection rules.

Finally, analysing HVAC systems requires a systems thinking approach, due to the complexity of the various (sub-)systems working together to provide a comfortable indoor climate. The impact of faults in the AHU on other components in the HVAC system, as well as the energy consumption of the building as a whole, need to be considered.

1.4. Thesis outline

The remainder of the thesis follows the following structure.

In the next chapter, a literature review and relevant theoretical background are presented. The literature review first gives an overview of general FDD for HVAC systems, followed by more specific methodologies for AHUs. The theoretical background section outlines related theoretical knowledge on AHUs, Bayesian network theory and building management systems (BMS).

In Chapter 3, the case study AHU and historical data are described.

Chapter 4 describes the Bayesian network-based fault diagnosis method that has been developed in this thesis. The faults, conditional probabilities and symptom detection rules of the DBN are outlined, as well as the data preprocessing steps.

In Chapter 5, the developed diagnosis method is validated with experiments conducted on the case study AHU. The experiment sensor data is diagnosed with different diagnosis periods and DBN structures to determine sensitivities.

The results of diagnosing the historical data are shown in Chapter 6. The faults diagnosed give insight into faults that occur during normal usage of the case study AHU.

The developed method for performing real-time diagnosis is shown in Chapter 7. This chapter shows the results of diagnosing data gathered through a data streaming platform.

In Chapter 8 a discussion is provided on the previously introduced results. The generalisability of the developed diagnostic method and sustainability implications are also described.

The thesis is concluded by Chapter 9, which contains answers to the formulated research questions, the main limitations of the conducted research and recommendations for future work.

2

Literature review and theoretical background

This chapter provides an overview of the current state-of-the-art literature and relevant background knowledge related to this thesis.

2.1. Literature review

In this section, the literature related to fault detection and diagnosis (FDD) of heating, ventilation and air-conditioning (HVAC) systems is described. Firstly, the various methodologies that can be employed for conducting FDD are presented. This is followed by a section that describes the literature that has utilised diagnostic Bayesian networks for FDD. Subsequently, the literature that diagnoses air-handling units (AHUs) is outlined. Finally, the section concludes with a description of the four symptoms three faults (4S3F) framework and an overview of the papers that have been published on this topic.

2.1.1. Fault diagnosis methodologies

Several methodologies can be applied to perform fault detection and diagnosis (FDD) for heating, ventilation and air-conditioning (HVAC) systems. The most established categorisation of these methodologies was introduced by Katipamula and Brambley in 2005 [27, 28]. They distinguished three categories: quantitative model-based, qualitative model-based, and process history-based methods. Quantitative model-based methodologies are based on physical models of the HVAC (sub-)system. The qualitative model methodologies include rule-based methods and qualitative physical models. The final category, process history-based methodologies, refers to models based on historical data.

In more recent literature review studies, the methods are often categorised as knowledge-based and data-driven instead [7, 37, 60]. The quantitative and qualitative model categories are both considered knowledge-based, and the process-history-based methodologies are referred to as data-driven methodologies. Additionally, the detection of faults is frequently distinguished from fault diagnosis.

Historically, mainly knowledge-based methods were used for FDD [27]. However, in the last two decades, most of the research has focused on researching novel data-driven methods. Between 1998 and 2018, 79% of the published papers on FDD applied data-driven methods [60].

Fault detection is also referred to as symptom detection and is used to check whether faults have occurred [47, 60]. This is followed by fault diagnosis, which is used to identify which faults are present based on the symptoms detected. Some data-driven FDD methods can be used to perform symptom detection and fault diagnosis simultaneously. Other methods can only be used for either symptom detection or fault diagnosis.

For example, regression can only be used for fault detection and Bayesian networks only for fault diagnosis.

Knowledge-based methods transfer expert knowledge into diagnostic models [60]. An example of a knowledge-based method for symptom detection is the rule-based method, which defines thresholds for sensor or control values corresponding to symptoms. Diagnostic Bayesian networks (DBNs) are an example of a knowledge-based fault diagnosis method. Examples of studies applying DBNs for fault diagnosis are given in Section 2.1.2

Knowledge-based methods have three main advantages: first, they require little fault data, particularly in physics-based modelling; second, they permit the system to be diagnosable even with uncertain input data; and third, the reasoning process and produced outcomes are interpretable. However, knowledge-based approaches have several limitations as well. Most importantly, they require expert input specific to the (sub)-system and are difficult to change or update without expert knowledge [37]. The accuracy of knowledge-based methods also tends to be lower than data-driven methods [60].

Data-driven FDD methods are not based on models, instead faults are detected or diagnosed by finding patterns in data. In contrast to knowledge-based methods, data-driven approaches do not require expert input. However, they have the disadvantage of requiring large amounts of data from the specific system being analysed [60]. Data-driven approaches are also commonly referred to as machine learning in the literature [7].

The data-driven methods can be further categorised into statistical and data-mining methods [37]. Principal component analysis (PCA) is an example of a statistical data-driven method. It requires fewer samples to find faults compared to data-mining approaches, however, also results in lower accuracy.

Data-mining methods can be further divided into supervised and unsupervised methods [37]. For supervised methods, an algorithm is trained on sensor data labelled faulty or fault-free. This is not done with unsupervised methods, which enables the algorithm to discover new knowledge and structure from the unlabeled data. Examples of unsupervised data-mining methods are clustering and association rule mining (ARM).

Supervised learning methods can also be further categorised into classification and regression methods, based on the output that is generated by the model [7]. Classification methods directly predict whether the sensor data is faulty or fault-free. Examples of algorithms used for classification are decision trees, Bayesian networks and support vector machine (SVM) models. Regression methods instead predict continuous variables, related to the operation status of the HVAC (sub-)system. These can be compared to the values during normal operation, to determine whether the system is operating with faults [7]. Support vector regressions (SVR) and neural networks are algorithms commonly used for regression-based FDD.

2.1.2. Diagnostic Bayesian networks

In this thesis, a diagnostic Bayesian network (DBN) is developed to perform fault diagnosis. This method was chosen based on successful applications in the literature, which are described in this section.

DBNs are Bayesian networks applied to the field of fault detection and diagnosis (FDD). Bayesian networks are probabilistic models that graphically represent causal relationships between variables as nodes and edges in a network [26]. In DBNs, faults and symptoms are used as variable nodes, enabling the calculation of the probabilities that faults occur based on detected symptoms. Each fault is assigned a probability of occurrence, independent of the presence of other faults. The symptoms are instead assigned probabilities related to the faults they are linked to in the network.

For example, a fault can be assigned a 10% probability of being present, while a symptom linked to this fault may be assigned an 80% probability of being present when the fault is present. From these probabilities, the DBN can infer the likelihood of the fault based on whether the symptom is present or absent. The underlying theory of DBNs is further described in Section 2.2.2.

DBNs have been widely applied to diagnose faults in a range of HVAC sub-systems, including AHUs [56, 61, 62], variable air volume (VAV) systems [58], heat pump systems [5], chillers [55] and variable refrigerant flow (VRF) systems [22]. The applicability of DBNs for various sub-systems of HVAC systems makes them well-suited for FDD of complete HVAC systems as well. The use of DBNs within the 4S3F framework is described in Section 2.1.4.

DBNs can be used in combination with both knowledge-based and data-driven methods. A recent example is the work by Wang et al., who successfully diagnosed faults with a DBN for an AHU with an ERW in the Netherlands [56]. In their study, symptoms were detected with regression, while the DBN structure was based on expert knowledge.

Data-driven methods can be used to find the probabilities or structure of the network [38, 57]. Furthermore, studies have also been conducted on extending the Bayesian network with dynamic conditional probabilities, which can change over time [43].

2.1.3. Air-handling unit diagnosis

The HVAC component that is analysed in this thesis is the air-handling unit (AHU), as it is responsible for a significant part of the energy consumption of the HVAC system. In this section, the relevant literature on AHU fault diagnosis is presented. AHU faults analysed in previous studies are shown, which were used to formulate faults included in the developed DBN shown in Chapter 4.

Yu et al. categorised common AHU faults into three types: equipment malfunctioning, actuator (dampers and valves), and sensor or controller faults [59]. These faults are summarised in Table 2.1. Distinguishing the faults in this way allows FDD to distinguish monitoring (sensor) faults from equipment faults.

Gunay et al. categorised AHU faults based on their impact on energy consumption and occupancy comfort [20]. They introduce several component faults that were not mentioned by Yu et al: an undersized coil, fan stuck at a constant rate, fouling duct and fouling or broken filter fault. AHU control faults are also described, such as incorrect airflow or temperature setpoints and incorrect scheduling of fans and coils. The components causing the highest increase in energy usage are a broken control valve, duct, filter or fan. Incorrect supply air settings for control settings can also lead to a high increase in energy usage.

A further category of faults that may affect AHUs is air leakage, which can occur in a number of ways [15]. First, external leakage can occur. In this case, air leaks from inside of the duct to outside, or vice versa. Second, internal leakage is also possible. This happens when air leaks between the ducts inside the AHU. This is most likely around the ERW where both airflows pass through. Third, air can leak past the filters, which decreases their functionality.

Chitkara simulated faults in AHUs for two Dutch buildings with the software EnergyPlus [9]. The faults were taken from the literature and analysed based on their impact on comfort and energy usage. It was found that control and sensor faults lead to higher energy usage than component degradation or fouling. The sensor measurements that had the highest impact on fault diagnosis were mass flow sensors. Unfortunately, these sensors are often not present in buildings, due to relatively high costs. Based on the energy impact analysis, the following faults were chosen for diagnosis in this study: a stuck or leaking control valve, fouled duct, fouled or broken filter, stuck or complete fan failure, sensor offset and inappropriate supply air set points.

Gopalan et al. also conducted a fault impact analysis study on AHU faults with EnergyPlus [17]. The results showed that fan failure, ERW failure, a stuck heating coil valve (HCV) and a stuck fan had the highest impact on energy usage during the winter. For the summer season failure of the fan had the highest impact.

AHUs have been diagnosed with multiple rule-based approaches [11, 40]. The most established work is the study by Schein et al., who provided a set of rules based on mass and energy balances for AHU diagnosis [44]. These rules are based on AHU systems with a mixing box instead of a heat recovery unit, so they cannot be applied to AHUs with an ERW.

Category	Component	Fault
Equipment	Fan	Pressure drop increase
		Supply and return fan failure
		Decreased motor efficiency
		Belt slippage
	Duct	Air leakage
	Coils	Decreased capacity due to fouling
Actuator	Dampers	Damper is stuck or in faulty position
		Air leakage in closed or open positions
	Coil valves	Valve is stuck, broken or operated to an incorrect position
		Air leakage in closed or open positions
Sensor	Temperature, humidity, flow rate or pressure sensors	Offset, drift or discrete sensor values
Controller	Motor modulation, sequence of coil valves, static pressure or zone temperature	Unstable response
	Flow difference	System stuck at fixed speed

Table 2.1: Faults in air handling units by Yu et al [59].

2.1.4. 4S3F framework

The four symptoms three faults (4S3F) framework is a recent approach using a DBN for fault detection [49]. The 4S3F framework is an architecture in which different symptom detection methods can be applied. The symptom detection method is coupled with a DBN to connect the identified symptoms to corresponding faults. The structure of the DBN in 4S3F is determined from HVAC system diagrams, for example, shown in Figure 2.1. Figure 2.2 shows an example of a DBN for the HVAC diagram shown in Figure 2.1. These probabilities need to be estimated or assumed to build the DBN, which is one of the limitations of the DBN method [61].

The main benefit of basing the framework on HVAC system diagrams is that it makes actual FDD implementation more accessible. The designers of the HVAC system can construct the diagnosis framework during the development of the HVAC system while determining the number of sensors needed and thresholds corresponding to faults in the system. This makes it possible for HVAC engineers to interpret results more easily from the framework and integrate an automated fault diagnosis into the HVAC system. Another benefit of the framework is that it is modular, and the diagnosis can be performed at different levels in the HVAC system. Taal divides the HVAC systems into five layers [48]. The highest level corresponds to the entire HVAC system. The second layer consists of generation, distribution and end-user systems. These layers can be further divided by aggregated systems, modules and components, such as the heat pump and heat pump for the third and fourth layers of a generation system. The modularity of 4S3F allows sub-systems to be diagnosed simultaneously, which can also be automated.

4S3F categorises four types of symptoms and three types of faults, diagnosed independently. The four symptoms are balance symptoms, based on deviations in balance equations; energy performance symptoms, based on energy performance metrics; operation state symptoms, based on operation states; and additional information, based on FDD systems integrated into components. The three fault types are component faults, relating to an incorrect amount of installed capacity or malfunctioning components; control faults, relating to incorrect settings for controllers; and model faults, relating to models estimating missing data. The relationships

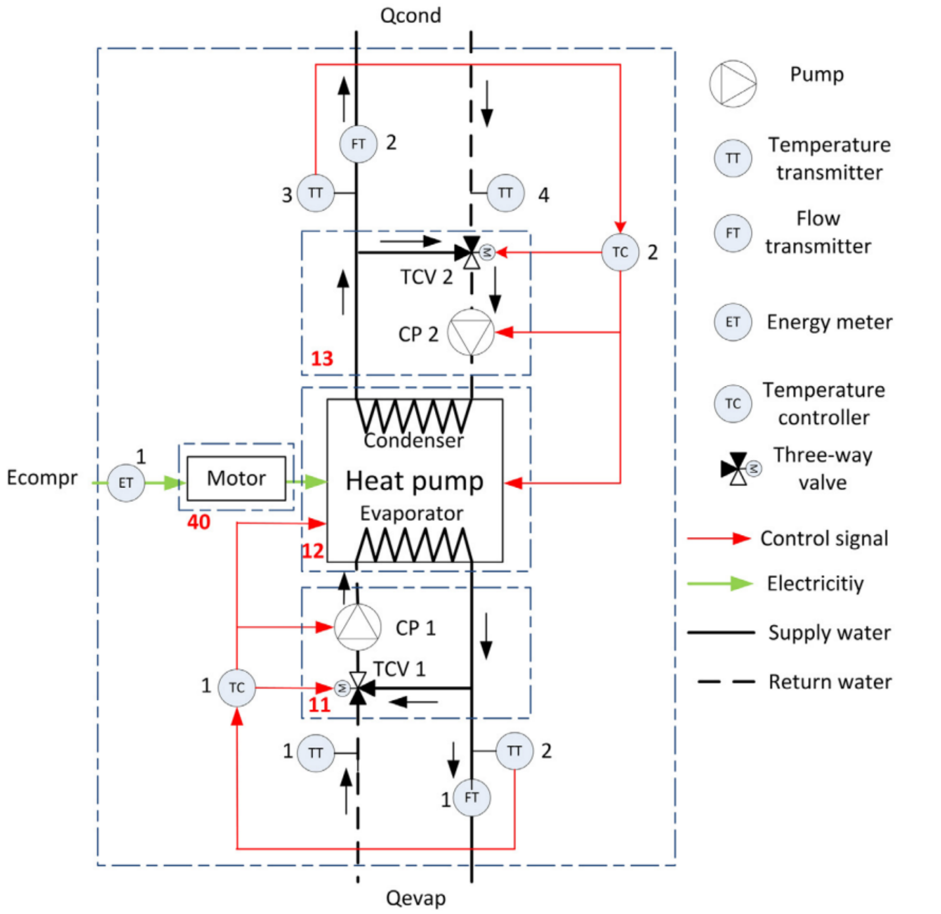


Figure 2.1: Example of a HVAC system analysed within the 4S3F framework [49].

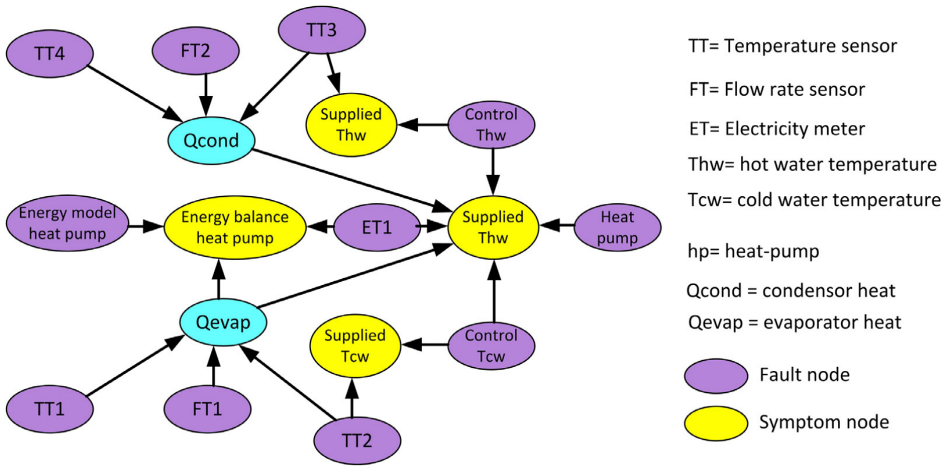


Figure 2.2: Example of a DBN within 4S3F [49].

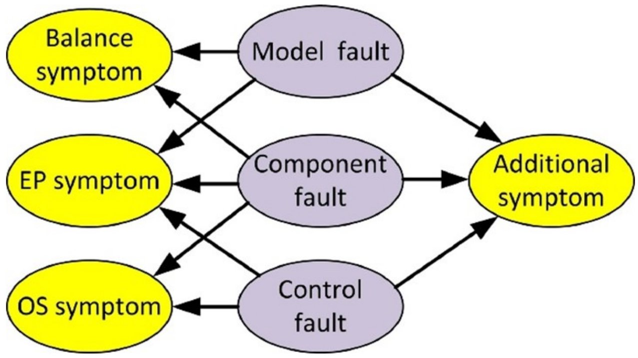


Figure 2.3: Relationship between types of symptoms and faults [49].

between the fault and symptom types are depicted in Figure 2.3.

Various literature has been published on the 4S3F framework after its introduction in 2018, showing it can successfully diagnose faults based on HVAC system diagrams. A paper demonstrating how a DBN can be set up from HVAC system diagrams was published by Taal et al. in 2019 [48], followed by an application of 4S3F to a ventilation system [45]. The symptom and fault detection parts of the 4S3F framework were further developed and applied to a thermal energy plant with a heat pump, gas boiler and an aquifer thermal energy storage system [46, 47]. The results of this case study showed 25% possible annual energy savings, even though the prior and conditional probabilities in the Bayesian network were estimated. Application of the framework on different time intervals is one area that needs further research [49], which this thesis will aim to contribute to.

Taal proposed several indicators for symptom detection in [47]. Different indicators are suggested for each of the four symptom types. The energy balance symptom indicators are based on heat- and mechanical efficiencies. These can be used to determine if the amount of output energy is significantly lower than the amount of input energy, which indicates energy waste. Three indicators for energy performance symptoms were introduced: performance factors, capacity indicators, and energy outliers. These indicators are based on reference values from HVAC design and combined with measured energy rates of components. The reference values are building-independent so these symptom detection rules can be applied to different buildings and HVAC systems. The performance of components can be calculated and compared to performance factors, such as the coefficient of performance (COP) and energy efficiency ratio (EER) specifications. Two types of indicators are found for operational state symptoms: control-based and design-based operational state (OS) indicators. Control-based indicators compare actual performance values to control values, such as supply air temperature and the control setpoint. Design-based indicators are not controlled in the system but are values assumed during the design of the HVAC system. Examples of these are the water supply temperatures. HVAC engineers can usually determine the set-point and design values directly from P&IDs. For additional symptoms, no generic indicators were introduced. Examples of the indicators for this category are maintenance information and user satisfaction. Table 2.2 lists all the symptom indicators introduced by Taal.

Symptom detection can be done over different periods, ranging from hours or days to months or annually. Each type of symptom in 4S3F requires a different timespan for detection. Energy balance symptoms were detected accurately daily in [47]. Daily detection is more challenging for energy performance symptoms, but this also depends on the specific system. All possible symptoms for different operating modes (e.g. heating and cooling) can be detected by incorporating all operation modes in a daily test run.

Wang et al. showed several possible DBNs by applying 4S3F to different AHU sensor configurations [57]. The DBNs contain the potential symptoms and faults of AHUs based on the 4S3F framework categories. The number of identifiable faults and symptoms depends on the availability of sensors in the system. The symptoms are divided into energy balance, energy performance and operation state symptoms, as included in

Symptom type	Indicator
Energy balance	Heat efficiency
Energy balance	Mechanical efficiency
Energy performance	Performance factor
Energy performance	Capacity indicator
Energy performance	Energy outliers
Operational state	Control-based
Operational state	Design-based

Table 2.2: Symptom indicators in 4S3F [47]

4S3F. Additional symptoms were not included. In their case study environment, the energy balance symptoms could be calculated only for the heating and cooling coils. However, in the best case, energy balance symptoms can also be found for the efficiency of the aggregated heating system, the energy balance of the heat-generating sub-system and an air pressure balance model. The amount of heat generated by the central heating system and supplied to the AHU must be measured to achieve this.

Li et al. have conducted a study using a similar approach to 4S3F [32], also applying DBNs for the FDD of HVAC systems. Their study looked only at the AHU component faults and did not consider energy and model faults, which are included in 4S3F.

2.2. Theoretical background

This section describes the relevant background knowledge on air-handling units, Bayesian network theory and building management systems (BMS).

2.2.1. Description of air-handling units

As mentioned in the introduction, Air handling units (AHUs) are often the target of FDD due to their high energy demand [7]. The unit can be situated in different parts of the building and is connected to the rest of the HVAC system with separate supply and return ducts. An AHU unit typically consists of fans, coils, filters, dampers, an energy recovery wheel or plate and sensor components. The placement of the components can differ per AHU and not all AHUs have an energy recovery wheel (also often called heat recovery wheel) or plate, which is further discussed at the end of this section.

The number of sensors present can also vary. At the bare minimum, supply temperature and supply/return pressure or airflow sensors are needed to make sure that the desired setpoint temperature and airflow are provided by the system. The damper components regulate the amount of airflow in the ducts, for example from outside into the building. A heating or cooling coil provides heating or cooling to the air by opening or closing a valve. For some AHUs, the cooling or heat-providing component is contained in the AHU, but this can also be provided by different units in other parts of the building.

AHUs can be separated into variable air volume (VAV) and constant air volume (CAV) types [20]. For VAV units the amount of airflow can be controlled, while CAV units deliver a constant air volume [59]. AHUs can also operate on different modes based on the outdoor temperature [44]. During cold weather, the AHU operates on heating mode and during warm weather on cooling mode. Additional modes can also be present for days when the outdoor temperature is closer to the supply temperature setpoint. The dampers, fans and heat recovery components can be controlled differently for each operation mode. For example, during the summer the fans can be used instead of the chiller when the outdoor temperature is close to the desired supply temperature. This reduces energy consumption.

Another significant distinction between AHUs is the method of heat recovery. For single-duct and dual-duct AHUs with a damper instead of a heat recovery unit, the return and supply airflows are mixed. This has become more controversial as a consequence of the COVID-19 pandemic [30]. AHUs with an energy recovery wheel or plate recover heat or cold from the return airflow to the supply flow without mixing the supply and return airflows. The airflows are only mixed when air leaks around the wheel or plate, which can happen unintentionally. As a consequence of the European Ecodesign Regulation 1253/2014 which has been in place since 2016, heat recovery has become mandatory in newly built AHUs in the EU [13]. The systems must also have an efficiency of at least 73%. As a result, AHUs with an energy recovery component will become increasingly common.

The differences between AHUs can lead to lower generalisability of fault detection methods. For instance, rule-based methods for FDD of AHUs without an ERW employ rules using the mixing temperature between the two ducts, which is absent in AHUs with an energy recovery component [44]. Because of this, established rule-based [44] or DBN-based [61] methods have to be adjusted before they can be applied to AHUs with an ERW. This is especially inconvenient as the ASHRAE datasets are frequently used in the literature, which are based on single-duct AHUs.

2.2.2. Bayesian network theory

In this thesis, a diagnostic Bayesian network (DBN) is used to diagnose faults based on detected symptoms. Diagnostic Bayesian networks are an application of Bayesian networks and are based on the same Bayesian probability theory. Therefore this section will briefly describe the necessary theoretical knowledge of Bayesian networks and probability theory.

Bayesian networks have a graph structure with nodes and edges in which each node represents an event. The nodes have multiple states relating to the outcome of the event. For binary nodes, there are only two states, true and false (also referred to as absent and present). The edges in the network are directional and indicate the probability of one event causing another. Each node in the network has either a conditional or prior probability, depending on whether an edge is directed towards it. If a node does not have any edges directed towards it, the event occurs independently of other events. In this case, the event has a prior probability. If a node has at least one edge directed to it, the probability depends on the state of the linked nodes. In this case, the probability of the event is referred to as a conditional probability. For example, consider a network with two nodes A and B and an edge from A to B. In this case, A has a prior probability, as it does not have any edges directed towards it. Event B has a conditional probability, which is described as the probability of B given A. The probability of event B being true thus depends on whether A is false or true. The table with all conditional probabilities of an event is called a conditional probability table (CPT). As Bayesian networks are built upon Bayesian probability theory, Bayes' theorem can be applied to infer probabilities. This formula gives the probability of event A given B, as shown in equation 2.1. These probabilities are referred to as posterior probabilities.

$$P(A|B) = \frac{P(B) \cdot P(B|A)}{P(A)} \tag{2.1}$$

Diagnostic Bayesian networks (DBNs) apply Bayesian networks to the field of FDD with the same mathematical principles of Bayesian networks. In DBNs, the nodes represent the faults and symptoms that can occur in (part of) an HVAC system. In DBNs, the nodes can also have more than two states. The edges in the DBN represent the probabilities of faults causing symptoms. The fault nodes have prior probabilities, indicating the probability that a fault is present independent of other faults. Symptom nodes have conditional probabilities, depending on the state of each connected fault. For FDD we want to know the posterior probabilities of the faults, which indicate the probabilities of faults being present given the detected symptoms. The posterior probabilities can be calculated with Bayes' theorem, based on estimated prior and conditional probabilities and detected symptom states.

Unfortunately, the conditional probability tables (CPTs) of symptom nodes grow exponentially in size for the number of faults they are linked to [61]. This is due to the need to define probabilities for every combination of fault node states. For every symptom node linked to n faults, the number of parameters in the CPT becomes 2^{n+1}. To address this issue, models can be used to generate CPTs based on fewer parameters. The method used in this thesis is the noisy-or model, which can be applied with binary fault and symptom nodes [12]. Binary nodes are used to simplify the network, leading to fewer conditional probabilities that need to be estimated.

The faults are assumed to be independent, meaning that fault probabilities are unchanged by the states of other faults. In the noisy-or model, the conditional probabilities of a symptom node are represented with parameters c_i, for every linked fault node i. The c_i refers to the probability of a fault causing a symptom to be detected. The formula for calculating the conditional probability is described in 2.2. The $I_(F)$ function describes the set of all i, where F_i is considered true for the conditional probability. This leads to a conditional probability of zero for the case where no faults are true. By using the leaky-or model, only the c_i has to be estimated for each symptom instead of all probabilities for every combination of fault nodes. This reduces the exponential number of parameters to a linear amount.

$$P(S|F) = 1 - \prod_{i \in I+(F)} (1 - c_i) \tag{2.2}$$

2.2.3. Building management systems

A building management system (BMS), or building automation system (BAS), is a platform used to control and monitor equipment in a building [36]. They are also used for controlling the HVAC system, which makes integration with a BMS necessary for performing automated FDD. There are several reasons for this. First, the controls of the BMS determine how the HVAC system behaves. For example, the operation schedule and temperature setpoints are set in the BMS. When these are set incorrectly, HVAC components may appear to be functioning incorrectly when this is caused by incorrect BMS controls. Therefore, the control signals sent to HVAC components from the BMS can be compared to intended designs to perform FDD. The second reason is that the BMS contains sensor data for symptom detection. These are sensors such as temperature or pressure sensors placed in the ducts or water pipes. Energy consumption data and component failure data can also be included in the BMS. Finally, a BMS can also contain functionality for creating alarms based on available sensor information. These can be used as an additional information source for performing FDD.

Modern BMS operate on an IP-based infrastructure, allowing operators remote control of multiple buildings [36]. BMS are also increasingly integrated with various Internet-of-Things (IoT) devices present in buildings.

3

Descriptive analysis of case study and historical data

In this chapter, the case study AHU of building 28 on the TU Delft campus is described. An overview is given of the various components within the AHU and their control mechanisms via the building management system (BMS) from Johnson Controls. The figures and explanations of the BMS are based on design documents and discussions with building operators.

3.1. Case study description

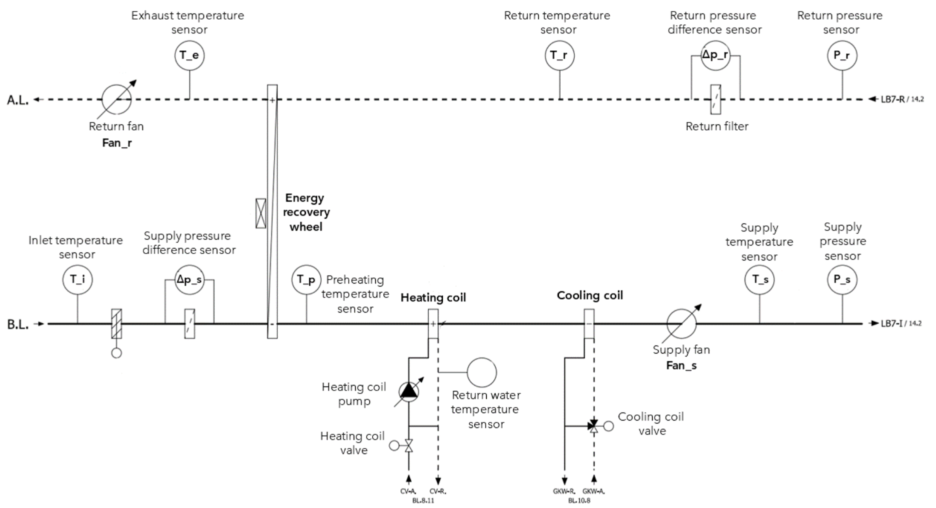


Figure 3.1: Component diagram of the case study AHU.

In Figure 3.1 the sensors and components of the analysed AHU are shown. The BMS control panel of the AHU is shown in Figure 3.2, in which the control signals of the components can also be found.

The AHU is operated by monitoring various sensors. There are three types of sensors used for this: temperature, pressure and pressure difference sensors. The air temperature sensors are located at the inlet (T_i), between the ERW and the heating coil (T_p) and at the end of the supply duct (T_s) on the supply side of the AHU. On the return side, they are located before the ERW (T_r) and between the ERW and the return fan (T_e). Pressure in the ducts is monitored by the two pressure sensors, located after the supply fan (p_s) and before the return filter (p_r). Two pressure difference sensors measure differences in pressure before and after the two

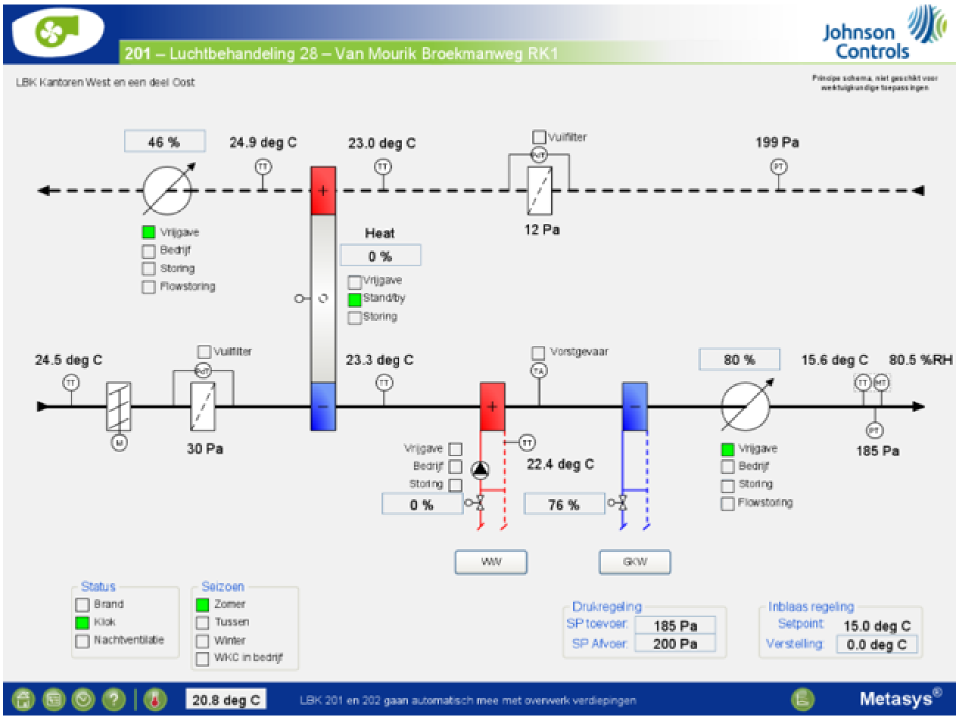
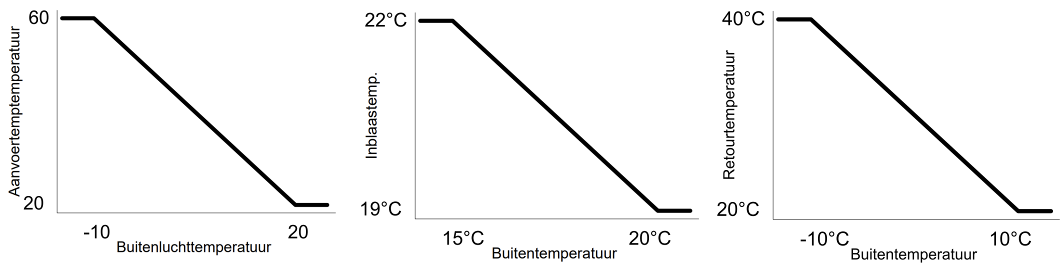


Figure 3.2: BMS control panel for the AHU in building 28, by Johnson Controls.



(a) Temperature setpoint of water supplied by the boiler to the AHU (y-axis), calculated based on outdoor temperature (x-axis).

(b) Temperature setpoint of air supplied by the AHU (y-axis), calculated based on outdoor temperature (x-axis).

(c) Heating coil return water temperature setpoint (y-axis), calculated based on outdoor temperature (x-axis).

Figure 3.3: Control rules of the AHU in the design document.

filters (Δp_s & Δp_r).

Heat and cold are provided to the AHU by two gas-powered boilers and a chiller located in the cellar of the building. Water is transferred from these units to the heating and cooling coils of the AHU, which heat or cool air supplied by the AHU. The required supply temperature of the boiler to the AHU is calculated based on the outdoor temperature, shown in Figure 3.3a. This is monitored by the BMS with a temperature sensor measuring the return water temperature at the heating coil. When the outdoor temperature is lower than 6 °C, the minimum heating coil return water temperature during startup is calculated based on the outdoor temperature. This is shown in Figure 3.3c. When this is not the case, no setpoint is used for the return water temperature. The heating coil also has a pump to retrieve water from the boiler, which the cooling coil does not have.

A temperature setpoint defines the desired temperature of the air supplied by the AHU. This temperature setpoint was set to 15 °C in the BMS in Figure 3.2. The setpoint temperature depends on the outdoor temperature, as shown in Figure 3.3b. The BMS uses the supply temperature sensor to monitor whether this

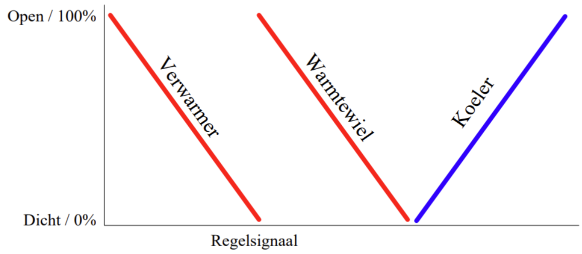


Figure 3.4: Order in which the coils and ERW are opened and closed from the design document. 'Verwarmer' refers to the HC, 'warmtewiel' to the ERW and 'koeler' to the CC. The y-axis shows the control signal, from 0% (closed) to 100% (open).

setpoint is being met. If the supply air temperature is below the setpoint, the ERW is activated, and if necessary, the heating coil valve (HCV) is also opened. When the supply air temperature is above the setpoint, the cooling coil valve (CCV) is opened further to reach the desired setpoint temperature. The valves control the amount of warm or cold water flowing past the coils, which in turn transfer heat or cold to the supplied air. In this manner, the ERW, HCV and CCV are operated based on the setpoint and measured supply temperature. The CCV is not operated during the winter, and the HCV is not used during the summer.

Ventilation of the AHU is provided by a set of fans located at the end of the supply and return ducts. This is the best configuration to minimise internal air leakage between the ducts, according to Eurovent, an association representing the European HVAC industry [15]. Pressure in the ducts is monitored with the pressure sensors. The AHU controls the fans to ensure that the measured pressure aligns with the supply and return pressure setpoints specified by the BMS. Both setpoints should always be set to 185 Pa during working hours, according to the design document of the AHU. The pressure setpoints are thus not dependent on outdoor temperature, moisture or CO_2 concentration.

Airflow into and out of the AHU is regulated with one supply damper and two return dampers. Dampers are valves or plates that can stop or regulate the flow of air. The dampers of the case study AHU are only controlled with signals to fully open or close. The supply damper is automatically opened by the AHU when the supply ventilator is turned on. The two return dampers are located on the east and west sides of the building and are opened by the BMS during working hours.

The AHU contains two air filters located before the ERW in the supply and return ducts. The filters are used to keep the components inside of the AHU clean and to clean the air supplied by the AHU. The pressure difference sensors are used as indicators that a filter needs to be replaced, as the pressure difference increases when dust accumulates in the filters.

An interesting component in the AHU for FDD is the energy recovery wheel (ERW), which has been studied less extensively in the literature. ERWs can be used to exchange heat or cold between in- and outgoing airflows. However, the ERW of the case study AHU is only turned on during working hours if the return temperature is higher than the outdoor temperature and if the supply fan is turned on. This means that the wheel is not used to recover cold, as the return temperature is then lower than the outdoor temperature. The corresponding control scheme is shown in Figure 3.4. As can be seen in the figure, the ERW is used to heat the air to the desired setpoint temperature, before the heating coil valve is opened. If the desired temperature cannot be reached by the ERW alone, the HCV is also increasingly opened. For cooling, the ERW is turned off and the cooling coil valve is operated to reach the desired setpoint temperature. The technical specification of the AHU states 49% efficiency for the supply airflow and 81% efficiency for the return airflow of the ERW.

The final components in the AHU are a moisture sensor and a frost sensor. The moisture sensor is used to measure moisture in the supplied air. The value is not used to control any components in the BMS and there are no alerts related to it. The frost sensor is used to measure frost on the heating coil. When the sensor signals that frost is present, the following actions are taken: the damper is closed, the supply fan is turned off,

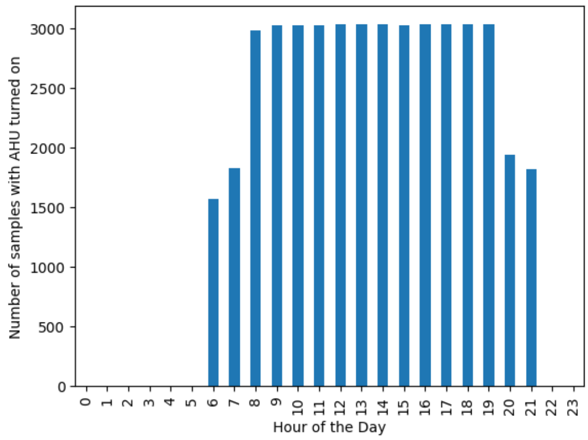


Figure 3.5: Operation schedule of the AHu.

the heating coil valve is fully opened and the heating coil pump is turned on. The system returns to normal settings when the sensor resets.

Besides the AHU, other units in the HVAC system, such as the boilers, chiller, radiators and ventilation, are also controlled with the BMS. These units are powered on and off along with the AHU on a shared daytime schedule. The BMS also features three seasonal modes: winter, summer and in-between seasons. The coils, boilers and chiller are blocked or unblocked based on the active season. Online weather data and functionality for creating alerts based on sensor data are also included. Finally, an event log containing the recent alert history and sensor data can be exported to Excel. These are not stored in the same manner as the sensor data, so they need to be exported separately. All sensor data is stored on local computers on the campus for approximately a month.

The following sensors were missing that could have been used to improve the diagnosis results: fan rotation speed, fan power consumption, airflow, return water temperature at the cooling coil and supply water temperature for both coils.

3.2. Historical data analysis

In this section, the historical data that was available for conducting the research is described.

The BMS includes an operation schedule for the entire building, according to which the AHU is activated. As shown in Figure 3.5, the HVAC system is mostly operational from 08:00 to 20:00. The AHU is turned off on weekends and several holidays, according to the schedule stored in the BMS. Based on this information, the dataset was filtered between 08:00 and 20:00 for days when the AHU was turned on.

Although sensor data was available from 2019 until 2023, only the data from 2022 onwards contained most of the necessary sensor data. The outdoor temperature data was stored from the 17th of November 2021 and the supply temperature setpoint from the 1st of August 2021. The return air temperature and pressure sensors were also added in 2021 and the supply pressure data was measured in 30-minute intervals until August 2021. Finally, the supply and return setpoints were available from the 2nd of August 2021. Because of these missing values, the research has been applied to the data from 2022 and 2023. During this period, the only sensor that was often unavailable was the return temperature sensor. Data for this sensor has only been stored from the 13th of September 2022, and from then only in 30-minute intervals. All other data was stored in 10-minute intervals. Figure 3.6 shows for which days data was missing, with a black square for every 10-minute interval.

Besides the sensor data, alert logs generated by the BMS were available for 2018 until 2022. The alerts appear infrequently, as in total only 12 alerts were signalled for the AHU in this period. The logs contain two types

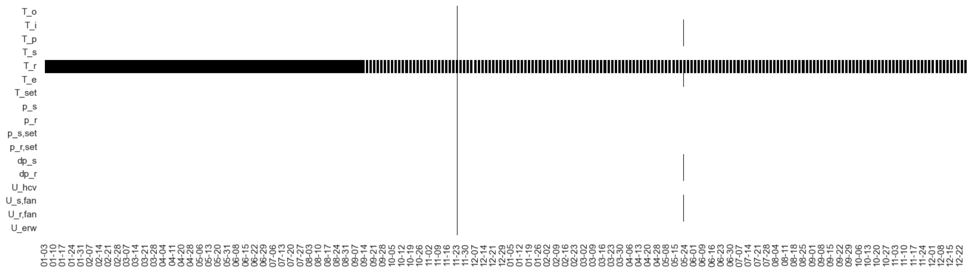


Figure 3.6: Available sensor data from 2022 until 2023. Missing data is shown in black for every 10-minute interval between 08:00 and 20:00.

of alerts, failure messages generated by components and alerts set by operators within the BMS. The alerts set in the BMS are flow alerts, generated by the BMS when the measured pressure is lower than 20 Pa, and fouling alerts, generated when the pressure difference at one of the filters is more than 250 Pa. Four flow alerts were present in the dataset, all occurring in 2020. Five failure alerts were for the ERW, of which three appeared in 2019, one in 2021 and one in 2022. Finally, a fouling alert appeared three times in 2020 for the supply filter.

Information on both gas consumption and electricity usage was available for the entire building, but not for the AHU specifically. Furthermore, building 28 is equipped with solar panels that contribute to its power supply, from which the amount of electricity produced was also available.

The distribution of the available sensor data is described with several boxplots for the filtered data. Figure 3.7 shows the distribution of the temperature sensors and temperature setpoint. The medians are shown with a green line and the average values with triangles.

Three interesting aspects are shown in the figure. First, the outdoor and inlet temperatures show a similar distribution, however, the inlet temperature is slightly higher on average. The outdoor temperature data comes from an online weather source in the BMS, while a temperature sensor in the BMS measures the inlet temperature. Consequently, one of the temperatures could be biased, or the difference is caused by different measurement locations. Additionally, the median of the preheating sensor is higher than the supply temperature sensor median, which is also unexpected since the Dutch climate requires heating more often than cooling. Therefore, the supply temperature sensor would be expected to be higher than the preheating sensor. Finally, the figure shows that the supply temperature setpoint is set lower than 19 °C for a large portion of the dataset, which contradicts the control document of the AHU. The setpoint can be seen to be set as low as 10 °C.

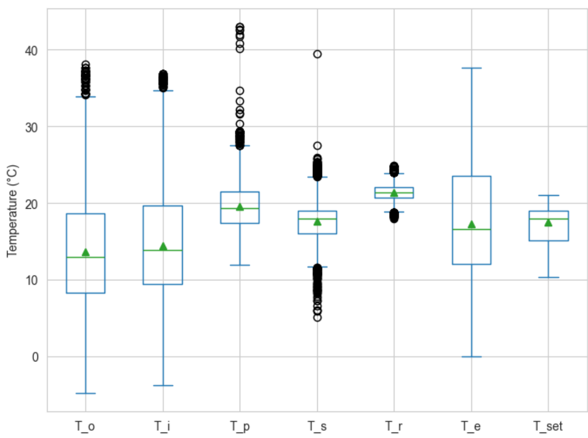


Figure 3.7: Temperature sensors and setpoint distributions.

Figure 3.8 shows the distribution of the pressure sensors, pressure difference sensors and pressure setpoints.

The boxes are relatively small, indicating that most of the data is centred around the median. As expected, the medians of the supply and return pressure sensors align with their respective setpoints. The supply pressure difference sensor has more outliers, while the values are also more concentrated around the median compared to the return sensor. The supply pressure setpoint is set at 185 Pa for all samples, the return pressure setpoint is set at 200 Pa in all but two samples. This is also not in line with the design document of the AHU, which stated that both fans should always be set to a setpoint of 185 Pa.

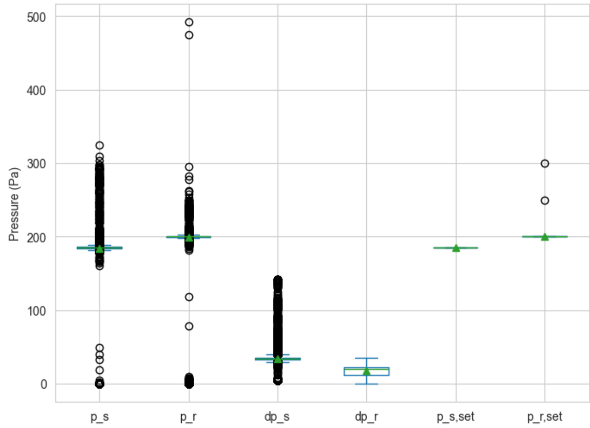


Figure 3.8: Pressure sensors and setpoint distributions.

The distributions of the control signals for the HCV, CCV, fans and ERW are shown in Figure 3.9. The signals sent to the components range from 0% to 100%. The distributions of the CCV and HCV control signals are relatively wide, due to the different heating and cooling needs depending on the outdoor weather. The cooling coil is used on fewer days than the heating coil, resulting in a 0% median. The medians of the fan control signals are expected to correspond to the control signal needed for adhering to the pressure setpoints. The boxes are relatively small, which should be the case as the pressure setpoints are always constant. The box of the ERW control reaches from 0% to 100%, as the ERW is operated to be turned off or on only, with 0% and 100% control signals. The binary control signals for the ERW can increase the difficulty of maintaining a stable supply temperature, as the ERW may need to be continuously turned off and on to stay at the desired setpoint temperature, instead of staying at a constant capacity like the valves. The median value of the ERW control lies at 100%, showing that the ERW is frequently turned on.

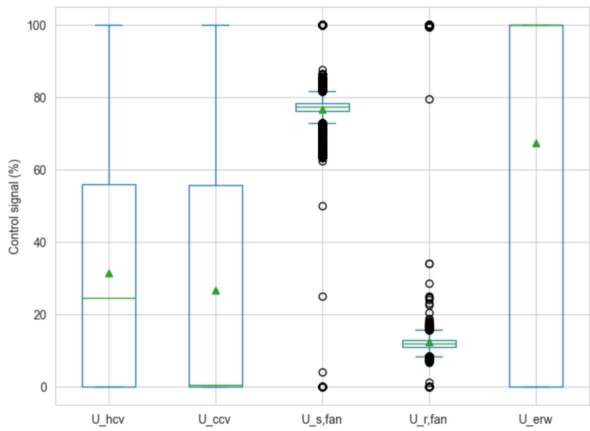


Figure 3.9: Control signal distributions

4

FDD model development

This chapter describes the implementation of the DBN for diagnosing AHU faults. First, an overview is given of AHU faults that can appear in AHUs, based on faults from the literature presented in Section 4.1. Then, the faults, symptom detection rules and conditional probabilities of the developed DBN are described. The chapter ends by describing the real-time implementation of the DBN.

4.1. AHU faults

The method that will be proposed in this thesis aims to diagnose faults in the AHU of building 28, while also being generalisable to other AHU systems. Therefore, the diagnosed faults should be common faults that are expected to apply to most AHUs in the Netherlands. Faults for AHUs without an ERW or with a preheating coil are thus excluded, as these are less common in the Netherlands. A list of these faults is presented in Table 4.1, based on AHU FDD literature presented in Chapter 2. Some general faults were out of the scope of this thesis, which was set to the operation of AHUs during working hours in the winter season. Therefore, faults have been excluded related to the cooling coil and incorrect control outside of operation hours.

Control faults are listed separately in Table 4.1 and refer to incorrect control signals from the BMS to the AHU. Faults for related equipment are combined for conciseness, such as faults for the fans at the supply and return ducts. Fouling faults occur when dirt or dust accumulates in a part of the AHU. Air leakage is categorised into internal leakage, where air leaks between the supply and return ducts, and external air leakage, where air leaks into or out of the AHU.

4.2. DBN description

This section outlines the developed DBN method. The DBN is shown in Figure 4.1. The following section describes and motivates which faults were chosen for diagnosis. Then, the chosen symptom detection rules and thresholds are described. Finally, the edges in the network are explained.

4.2.1. Fault diagnosis

Based on the faults found in the literature, 27 faults are identified in Table 4.2 that can be diagnosed for AHUs. The fault types are also shown, categorised into component and control faults as described in the 4S3F framework. Most faults are of the component fault type, which are faults related to the physical malfunctioning of components. The second fault type is the control fault type, which is considered as the BMS sending incorrect control signals to AHU components such as the fans, HCV or ERW. These faults can be diagnosed by checking whether the control signals correspond to the intended design of the control schemes. The final fault category from 4S3F, model faults, is not included as no efficiency calculations or value prediction methods are used for symptom detection. This is further discussed in Section 4.2.2.

Component	Fault description
Heating coil	Valve stuck in incorrect position; valve leakage; unstable valve control; undersized heating coil; fouling heating coil.
Fan	Efficiency degradation; stuck at constant speed, increased pressure drop
Duct	Internal or external air leakage; fouling
Filter	Leakage past the filter; fouling
ERW	Complete failure; stuck at constant speed; air leakage
Damper	Stuck in incorrect position; air leakage
Pressure, temperature, airflow, humidity and CO_2 sensors	Bias, drift or discrete values
Control	**Fault description**
Setpoint	Temperature setpoint incorrect; airflow setpoint incorrect; pressure setpoint incorrect
Component control	ERW control incorrect; HCV control incorrect; fan control incorrect; damper control incorrect
BMS control	Operating schedule incorrect; control sequencing of valves and ERW incorrect

Table 4.1: General AHU faults found in literature [8, 20, 59]

The following paragraphs describe nine types of faults, which were chosen not to be included in the DBN.

First, faults related to CO_2 and moisture sensors, as these are not used for controlling the AHU and are not used for diagnosing other components in DBNs in the literature [61, 62].

Second, airflow sensor-related faults, as the AHU uses static pressure sensors instead of airflow sensors. Corresponding control symptoms have also been removed.

Third, the efficiency degradation of the fans. This can be measured by comparing the power consumption or rotation speed of the fans to the airflow output [62]. However, the power consumption and rotation speed of the fans are both not available in the case study AHU.

Fourth, the bias fault for the supply/return pressure sensors and the two pressure difference sensors. These cannot be diagnosed because the pressure values cannot be determined from airflow or power consumption data.

Fifth, fouling of the HCV and undersized HC capacity are not included due to a missing supply water temperature sensor. This sensor is needed for calculating the expected heat transfer of the heating coil.

Sixth, damper leakage is not included as the diagnosis is done during working hours of the AHU. During these hours the damper should always be opened.

Seventh, the incorrect operating schedule fault is also not included as only diagnosis during working hours of the AHU is considered.

Eight, exhaust damper failure is excluded as the exhaust dampers are considered out of the scope of AHU diagnosis. This is because the exhaust dampers are situated in different parts of the building and are therefore considered part of the ventilation system.

Component	Fault	Description
Fan$_s$, fan$_r$	Failure	The supply or return fan is stuck at a certain speed or fails completely.
Filter$_s$, filter$_r$	Leakage	Air is leaking past the supply/return filter.
	Fouling	The supply/return duct or filter is filled with dirt/-dust.
Heating coil	Failure	The heating coil valve is stuck at a certain position.
	Leakage	The heating coil heats the air when the valve is closed.
ERW	Failure	The energy recovery wheel is stuck at a certain rotation speed or fails completely.
Damper$_s$	Stuck closed	The supply damper is stuck closed.
P$_s$ sensor, P$_r$ sensor	Failure	The supply/return pressure sensor shows outliers or data is not available.
T$_i$, T$_p$, T$_s$, T$_{re}$ combined	Bias	A temperature sensor is biased positively or negatively
T$_i$, T$_p$, T$_s$, T$_r$ and T$_e$	Failure	A temperature sensor stores no data or shows outliers.
Control	**Fault**	**Description**
Temperature setpoint	Setpoint set incorrectly	The AHU supply temperature setpoint is higher or lower than designed.
P$_s$ setpoint, P$_r$ setpoint	Setpoint set incorrectly	The AHU supply/return pressure setpoint is higher or lower than designed.
ERW control	Incorrect control signal	The ERW control signal should be increased or decreased based on the supply temperature and setpoint.
HCV control	Incorrect control signal	The heating coil valve signal is incorrect.
Fan$_s$ control, fan$_r$ control	Incorrect control signal	The supply/return fan control signal is incorrect.

Table 4.2: Faults diagnosed with the developed DBN.

Finally, duct fouling and leakages are not included in the DBN for building 28, as they have a small impact on energy usage and are among the hardest faults to detect [9]. However, filter fouling has been included which can lead to duct fouling. Fouling of the duct leads to filter fouling when the dirt from the ducts is blown into the filters.

As mentioned in the previous paragraphs, several sensors or control values were absent in the AHU, because of which some faults were chosen not to be included in the DBN. The missing sensors and control signals were the following: damper control signals; fan rotation speed sensors; fan power consumption; supply and return airflow sensors; cooling coil return water temperature; and heating and cooling coil supply water temperature sensors. For other AHUs, if the sensors are available the DBN structure could be altered in the following way. The damper control signals could be used to check for incorrect control of the dampers, separating the damper fault into a damper and component fault. High power consumption of the fans can be added as symptoms linking to a new low fan efficiency fault, based on the designed fan efficiency of the AHU. Low airflow and fan rotation speed symptoms could be added as symptoms linked to both fan control and fan failure. With power consumption, airflow or fan rotation speed sensors available, pressure sensor biases could also be diagnosed. The pressure bias fault could be linked to a symptom that compares the measured power, airflow or fan rotation speed to the measured pressure. Finally, if the water temperature sensors are present, the coil efficiencies can be calculated and added as a symptom. These would also be linked to added low HC or CC efficiency faults.

Several of the faults presented in Table 4.1 have been combined to lower the number of nodes in the DBN. These are the failure and stuck component faults for the ERW, HCV and fans. Drift and discrete values faults have been combined with bias faults, as these can all be detected by comparing the values of different sensors. For the temperature sensors on the supply side, the failure and bias faults have been combined. The control sequencing fault for the HCV and ERW is included in the HC control and ERW control faults. The supply damper control fault is combined with the failure fault, as it is mechanically operated inside the AHU. Therefore no control signal is sent to the damper from the BMS.

All prior fault absent probabilities are set to 98% for components and 95% for control faults, which are also used in the DBNs introduced by Taal et al [46]. Thus a 2% probability is expected of any component fault being present and a 5% probability of any control fault, independent of any other faults or symptoms.

The posterior probabilities of the faults are the probabilities of faults being present given the detected symptoms. These probabilities are inferred by inputting the detecting symptoms into the DBN. A fault is considered to be detected if the probability is at least 60%. This value was chosen as it lies between the 70% threshold used by Zhao et al. [62] and the 30% threshold used by Taal et al [45].

The implementation of the DBN contains multiple features for performing real-time fault diagnosis. The first feature is a configurable diagnosis period, which is set as a number of 10-minute samples. As the DBN is formulated for performing diagnosis of the AHU during operation, this period ranges from one 10-minute sample up to all samples during which the AHU was operated. Based on the historical data, the AHU was shown to be operated on a 12-hour schedule, which results in a maximum diagnosis period of 72 samples. When the timespan has been determined, fault diagnosis is applied by detecting the symptoms for the chosen timespan. Following that, the DBN is constructed and the faults are determined by inferring the probabilities with the detected symptoms as evidence.

Additionally, two features have been added to improve the diagnosis results for longer detection periods. First, symptoms need to be detected at least a certain number of times before they are considered detected. Second, each symptom needs to be detected several times consecutively. The chosen thresholds are described in the diagnosis setup sections in Chapter 5 and Chapter 6. The first threshold will be referred to as the total number of symptoms threshold and the second as the consecutive symptom threshold.

When combined, the thresholds determine that symptoms are only considered detected if they appear in multiple samples consecutively a certain number of times. For example, with a total number of symptoms threshold of three and a consecutive symptoms threshold of two, for at least three samples the symptom also has to be detected in the previous sample to be considered detected. The used threshold values are listed in the experimental setup of Chapters 5 and 6.

4.2.2. Symptom rules

A knowledge-based approach is used for symptom detection. This implies that the DBN does not need historical data to perform symptom detection by applying methods such as regression or machine learning. Instead, rules are determined based on knowledge of how AHU systems are supposed to operate during fault-free operation. This approach is similar to symptom detection rules presented by Taal et al. and the APAR rules [44, 45]. Three of the symptom detection rules used by Chitkara are also similar because they also do not require historical data [8]. These rules are the comparisons between the measured and setpoint air temperatures. Several rules applied by Zhao et al. are also knowledge-based [61, 62].

The symptom detection rules used in the DBN are listed in Table 4.4. Symptoms and faults in the introduced DBN have only two states, present or absent. This simplifies the construction of the DBN, by needing only

Threshold	ε_1	ε_2	ε_3	ε_4	ε_{5s}	ε_{5r}	ε_6	ε_7	ε_8	ε_9
Value	5 Pa	5 Pa	1 Pa	239 Pa	174 Pa	5 Pa	1 °C	1 °C	1 °C	5
Threshold	ε_{10}	ε_{11}	ε_{12}	ε_{13}	ε_{14}	ε_{15}	ε_{16}	ε_{17}	ε_{18}	ε_{19}
Value	2 °C	1 °C	1 °C	0.49	-25 °C	50 °C	0 Pa	500 Pa	19 °C	22 °C

Table 4.3: Thresholds values used for symptom detection.

one parameter for each conditional probability between a fault and symptom. It also helps in making the network more interpretable. Besides available sensor data, the rules also use additional data provided by the BMS, mentioned in Chapter 3.1. These are the various alerts generated by malfunctioning components, which are the fans, HC and ERW, as well as alerts generated by the BMS related to the airflow. The outdoor temperature values stored in the BMS come from online weather data and are referred to as T_o in the DBN. This data is assumed to be correct, and therefore no outdoor temperature bias node was added.

The symptom rules were formulated to have strong correlations to the chosen faults. In this way, a DBN is created with fewer edges, leading to smaller CPTs (conditional probability tables), faster computation and a more interpretable network. The ε thresholds determine how much the values can deviate from the expected values. Increasing the thresholds leads to more detected symptoms and decreasing the threshold values does the opposite. They therefore have a significant effect in determining which symptoms are detected. In general, the thresholds were chosen to be relatively large, leaving some room for error. The chosen ε values are listed in Table 4.3. An explanation of the chosen values is given at the end of this section.

Most of the symptoms are operational state (OS) symptoms from 4S3F. These are based on measured values, for example from temperature and pressure sensors. The only balance symptoms are the temperature bias symptoms, which compare the temperatures at different locations in the AHU. No air pressure balance model could be constructed, as no pressure sensors are available at the inlet and outlet of the air ducts. In addition, water temperature sensors were not present to detect aggregated or sub-system energy balance symptoms, as suggested by Wang et al [57].

Two energy performance (EP) symptoms are used, for detecting low ERW and HC efficiency. There are no additional EP symptoms, as no temperature or pressure balance model is created. Additional symptoms are not considered, which are optional for further improving the diagnosis results.

The following sections describe all of the formulated symptom rules, categorised by the sensors compared in the symptom detection rule.

Temperature-related symptoms

Symptom S1 detects whether the damper is unintentionally closed, by comparing the inlet temperature sensor to the outdoor temperature data. The temperatures should be similar, as the air is not heated before the inlet sensor.

Symptom S16 is similar to S1, as both compare the outdoor and inlet temperatures. The threshold for symptom S16 is smaller compared to S1 however, to detect whether the inlet sensor is biased. This rule also contains thresholds for detecting outliers or missing values for the inlet sensor, to decrease the size of the network. This symptom rule is based on the assumption that the outdoor temperature data is correct, allowing the inlet temperature to be compared to the outdoor temperature for detecting a bias.

Also on the supply side, S17 detects failures or biases of the preheating temperature sensor. By detecting both failures and missing data, the number of symptoms in the AHU is decreased. Symptom S17 is detected if the difference between the preheating temperature and the inlet temperature is greater than 1 °C when the ERW is off. If the temperatures are expected to be equal, as otherwise one of the sensors could be biased.

Symptom S18 is similar, but instead compares the temperatures before and after the heating coil. The absolute difference between these temperatures should be small when the heating coil valve is closed. When the difference is larger than 1 °C, while the HCV and CC control signal are 0%, the symptom is detected. Additionally, the symptom is detected if missing values or outliers are detected.

If there is a temperature difference before and after the heating coil, it could be either caused by biased temperature sensors, or a leaking heating coil valve. To distinguish these faults, symptom S26 was added, which also compares the temperatures with the same threshold of 1 °C. However, for this symptom, the difference is taken as the supply temperature minus the preheating temperature, instead of the absolute temperature.

On the return side, symptoms S19 and S20 detect failures and outliers for the return and exhaust temperature sensors, while S21 is used for detecting a bias between the return and exhaust temperature sensors. Symptom S21 is detected if the ERW is turned off and the difference between the return and exhaust temperature sensors is greater than ε_7. The bias fault for the return and exhaust temperature sensors has been combined because the sensor values cannot be compared to any other sensors, which was the case for the supply-side temperature sensors. Therefore only diagnosis of a bias between both sensors can be diagnosed, but not for either specifically.

Pressure-related symptoms

There are no symptoms for detecting whether the pressure sensors are biased, due to lacking sensors for the case study. However, failures are detected for the pressure sensors in symptoms S30 and S31, by checking for outliers and missing data. For the pressure difference sensors, symptoms S32 and S33 detect outliers or missing data.

Setpoint-related symptoms

Three OS symptoms (S4, S7 and S23) compare measured values to the setpoints. This type of symptom is also often present in DBNs published in the literature, as they indicate that the AHU is failing to deliver the air at the required rate or temperature [45, 62]. Symptom S23 is detected when the absolute difference between the supply temperature sensor and the setpoint is larger than threshold ε_9. Symptoms S4 and S7 are detected when this is the case for the supply and return pressure respectively, using threshold ε_3.

Incorrect setpoints are detected with OS symptoms S8 for the supply pressure, S9 for the return pressure and S34 for the supply temperature setpoint. S8 is detected when the supply setpoint is set at least 5 Pa differently than 185 Pa and S9 is detected when the return setpoint is set more than 5 Pa differently from 200 Pa.

As stated in Section 3.2, 185 Pa and 200 Pa are used based on the historical data. Symptom S34 is detected when the setpoint is set below 19 °C or above 22 °C, which should not happen according to the design document of the AHU shown in Figure 3.3b. Another approach for this symptom could have been to calculate the intended setpoint value based on the linear relationship shown in Figure 3.3b, but this was not done to allow for some deviation from the design document.

Component control-related symptoms

Incorrect control signals are detected for the fans, heating coil valve and energy recovery wheel. These symptom rules are the most sophisticated, combining multiple statements. All of these symptoms are detected when the control signal is less than 0% or greater than 100%. Additionally, the setpoint temperature or pressure is used to check whether the control signal should be set lower or higher. For the supply and return fans symptoms S3 and S6 are formulated. Their symptom detection rules state that the control signal should be increased when the respective pressure setpoint is not being met, while the control signal is not at the maximum value (100%). The opposite case is detected if the control signal is not set to 0% even though the current measured pressure is higher than the setpoint value. The threshold used for deviation from the setpoint is ε_2. Symptoms S22 and S27 apply a similar rule, detecting a control symptom when the difference between the measured supply temperature and setpoint becomes larger than thresholds ε_8 and $\varepsilon_1 2$. Both symptoms are also detected if the signals do not follow the correct sequencing. The ERW should always run at 100% capacity if the heating coil valve is opened, otherwise, the heating coil is used to heat the air while heat recovery could be applied. Therefore the control signal for the ERW should not be lower than 100% while the HCV control signal is above 0%. In this scenario, both of the control signals are incorrect, and therefore the rule is added for both symptoms.

Efficiency-related symptoms

Two symptoms detect low efficiencies, S25 for the heating coil and S28 for the ERW. Because the return water temperature is missing, the efficiency cannot be calculated directly from the difference between the water before and after the heating coil. Therefore, a different method was used as an indication of the efficiency. The symptom rule calculates the difference between the preheating and supply temperatures and compares this to the HCV control signal. Threshold ε_{10} is multiplied with the control signal for the minimum required temperature difference. The threshold is therefore based on the expected minimum temperature difference when the valve is fully opened. Symptom S28 detects low efficiency for the ERW, by comparing the temperatures before and after the ERW. The efficiency calculation is based on a formula for an ideal heat recovery system, where all of the heat is transferred [6]. The efficiency of ERWs can be calculated in two different ways, either by comparing the difference in airflow temperatures in the supply or return duct.

S28 uses the return side instead of the supply side for calculating the efficiency. This is done because the heating coil on the supply side could influence the measured preheating temperature. A side benefit of using the return instead of supply temperature sensors is that the supply efficiencies of ERWs are often overestimated, while the return efficiencies are underestimated [6]. Therefore, the listed return efficiency of the ERW is better used as a minimum expected efficiency threshold. However, for this case study specifically using the return side is less convenient since the return temperature sensor data is stored less often than the preheating sensor. This decreases the number of samples for which this symptom can be detected.

Alert-related symptoms

Several of the symptoms are related to BMS alerts. Symptoms S2 and S5 are detected when the BMS shows a failure alert message related to one of the fans. Symptoms S24 and S29 are also linked to alerts generated by components, S24 links to the heating coil pump failure alert and S29 to the ERW failure alert. The alerts of symptoms S11 and S13 are created by the BMS, set by the building operators to turn on when the pressure difference at one of the filters exceeds 250 Pa.

Filter-related symptoms

Four symptoms have been added to detect leakage and fouling of the supply and return filters. Symptoms S10 and S12 detect filter fouling by comparing the supply or return pressure difference to threshold ε_{5_s} or ε_{5_s}. Filter leakage is detected with low-pressure difference symptoms S14 and S15. The symptoms are detected if the control signal of the fan is at least 5% and the pressure difference is less than the expected threshold ε_6.

Epsilon threshold values

As mentioned at the beginning of this section, the threshold values shown in Table 4.3 have a strong influence on how often symptoms are detected. The thresholds for bias detection and deviation from the temperature setpoint have all been set to 1 °C for consistency. For deviation from the pressure setpoint, the threshold was set to 5 Pa. Where possible, data from the AHU was used for determining the values.

For the filters, the expected minimum and maximum pressure differences were available in the design documents. The minimum values were not used, however, as they were higher than the measured pressure differences shown in Figure 3.8. The maximum resistances before needing replacement were used for thresholds ε_{5_s} and ε_{5_r}. The return side efficiency of the ERW was used to set threshold ε_{13}. The values of other thresholds were based on estimations.

4.2.3. Conditional probabilities

The developed DBN, presented in Figure 4.1, shows all the relationships between the symptoms and faults. Some of the faults and symptoms that occur in both the supply and return sides of the AHU have been combined in the figure. This is indicated with a subscript 's/r'. For combined fault nodes, the structure of edges to symptom nodes is different depending on whether the symptom node is also combined. If the symptom node is also combined, the supply side fault is linked to the supply side symptom and the return side fault is connected to the return side symptom. For example, the top-left $T_{r/e}$ failure fault represents the $T_{r/e}$ fault connected to symptom S19 and the T_e fault connected to symptom S20. If the symptom node contains one symptom, both faults are connected to it. An example of this is the connection from the $T_{r/e}$ failure fault to symptom S21, representing that the T_r fault and T_e fault are both connected to symptom S21. The damper is connected only to symptoms S14 and S4 because these are both supply-side symptoms.

All symptoms and faults in the DBN are connected, there are no disjoint graphs. However, the DBN can be loosely divided into temperature-related faults and symptoms at the top, and pressure-related symptoms and faults at the bottom. There are few edges between these sides, with three exceptions. These are edges from fan failure, fan control and damper faults, connecting from the bottom to the top part of the DBN. The damper failure fault is connected to symptom S1 because a closed damper would cause the inlet temperature to deviate from the outdoor temperature. The fan failure and control faults are connected to the heating coil and ERW efficiency because the efficiencies assume sufficient airflow.

The other relationships between symptoms and faults were determined as follows. The sensor faults are connected to symptoms where their values are used in the symptom detection rule, such as the supply temperature for calculating the deviation from the setpoint in symptom S23. The only exception is the connection between the pressure difference faults and the filter alerts, which is because the filter alerts also rely on the pressure difference data. The component failure faults of the fans, HC and ERW are linked to corresponding alert, efficiency and setpoint comparison symptoms. The HC leakage fault is connected to symptoms S18 and S26, which calculate the difference between the temperature before and after the heating coil. Failure of the supply damper can cause a difference between the inlet and outdoor temperatures, or a lower supply pressure (difference). Therefore this fault has been linked to symptoms S1, S4 and S14. Filter leakage has little influence on measured data, and is therefore connected only to the low pressure difference symptoms (S14 and S15). The filter fouling fault is connected to both the alert (S11 and S13) symptoms and the high pressure difference symptoms (S10 and S12), which are both detected with a high pressure difference.

For the control faults, all related temperature or pressure faults were connected. For the HC and ERW control faults, these are the symptoms that check their control values (S22 and S27), as well as a deviation between the measured supply temperature and setpoint (S23). The fan control faults are linked to the fan control symptoms (S3 and S6), as well as the symptoms comparing the measured pressure and setpoint (S4 and S7). As mentioned above, the fan control symptoms are also connected to the efficiency symptoms (S25 and S28). The supply temperature setpoint fault is connected to symptoms S34 and S23 because both symptoms use the setpoint value. This reasoning was also followed for the pressure setpoint faults.

The conditional probabilities of the network have been simplified into two types: strong and weak links. Black edges represent strong conditional probabilities of 90% and grey edges weak conditional probabilities of 30%. Theoretically, separate conditional probabilities can be defined for every edge in the network, but this becomes cumbersome in practice. Furthermore, Taal et al. have shown that the chosen absolute conditional probability values are less important than the relative values [45].

For each component fault, the conditional probability of the related alert symptom was set to strong. If a fault had no related alert, the most likely symptom was considered strong. This approach was made possible because symptoms were formulated to correspond to the chosen faults. The ERW and HC efficiency symptoms were also considered strong conditional probabilities for the ERW and HC failure faults.

Symptom S23 is connected to a relatively large number of faults, as it can be caused by multiple control, component or sensor faults. This is also the case for symptoms S4 and S7, which are detected when the measured pressure is different from the setpoint. Because of this, they have no strong conditional probability edges, as the number of possible causes is too large to be certain which fault likely caused it.

The entire CPT for each symptom node is constructed with the leaky-or model based on the conditional probabilities, as described in section 2.2.2. The CPT of a symptom node contains probabilities for each combination of fault states that are connected to the node.

4.3. Data preprocessing

The preprocessing of the different datasets that were used for diagnosis is described in this section.

Fault diagnosis was applied to three different datasets. First, a comma-separated values (CSV) file with the experiment sensor data. Secondly, a CSV file containing the historical data of 2022 and 2023. Finally, the pandas DataFrame that was exported from the Kafka cluster. While the datasets were preprocessed similarly, there were some differences.

For the historical data from 2022 and 2023, the two CSV files were first combined and then processed in several steps. First, alerts from the alert log were added to the dataset, aligned with the corresponding 10-minute intervals. Next, the data was indexed by date and time. The columns necessary for analysis were then renamed and unnecessary columns were removed to reduce the DataFrame's size. Additionally, strings containing commas were converted to floats.

Compared to the historical data, the experiment data contained different column names and time formatting. Outdoor temperature data and setpoint data were not present in the dataset, so these were added manually. This is further described in Section 5.2. No alerts were signalled during the experiments, but the columns were still added so the same DBN structure could be applied. Finally, the data was converted to 10-min samples. The CSV file contained rows for every 5 minutes, although most of the data was stored in 10-minute intervals.

The real-time data exported from Kafka also consisted of different column names compared to the historical and experimental datasets. In addition, this data was stored in Coordinated Universal Time (UTC) format, which needed to be converted to the local 'Europe/Amsterdam' time zone. Alerts were not yet part of the Kafka environment, so these were also added to the DataFrame manually as columns of zeros.

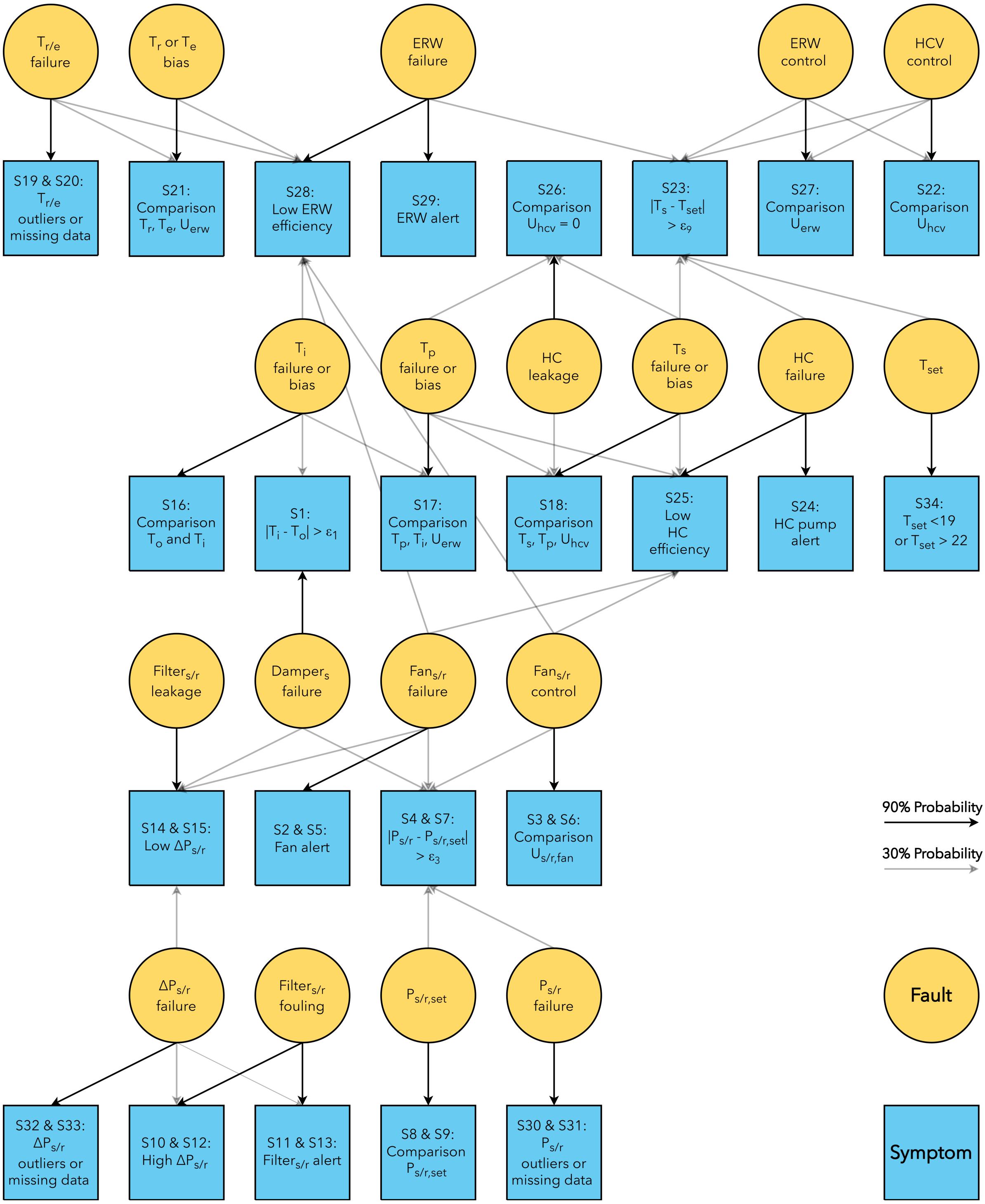


Figure 4.1: DBN developed for the case study AHU.

	Symptom	Rule
S_1	Comparison T_i and T_o	$\lvert T_i - T_o \rvert > \varepsilon_1$
S_2	Supply fan alert is present	$A_{s,fan} = 1$
S_3	Comparison $U_{s,fan}$, $p_{s,set}$ and p_s	$U_{s,fan} < 0\%$ or $U_{s,fan} > 100\%$ or $(U_{s,fan} < 100\%$ and $p_s < p_{s,set} - \varepsilon_2)$ or $(U_{s,fan} > 0\%$ and $p_s > p_{s,set} + \varepsilon_2)$
S_4	Difference $p_{s,set}$ and p_s	$\lvert p_s - p_{s,set} \rvert > \varepsilon_3$
S_5	Return fan alert is present	$A_{r,fan} = 1$
S_6	Comparison $U_{r,fan}$, $p_{r,set}$ and p_r	$U_{r,fan} < 0\%$ or $U_{r,fan} > 100\%$ or $(U_{r,fan} < 100\%$ and $p_r < p_{r,set} - \varepsilon_2)$ or $(U_{r,fan} > 0\%$ and $p_r > p_{r,set} + \varepsilon_2)$
S_7	Difference $p_{r,set}$ and p_r	$\lvert p_r - p_{r,set} \rvert > \varepsilon_3$
S_8	Comparison $p_{s,set}$	$\lvert p_{s,set} - 185 \rvert > \varepsilon_4$
S_9	Comparison $p_{r,set}$	$\lvert p_{r,set} - 200 \rvert > \varepsilon_4$
S_{10}	High Δp_s	$\Delta p_s > \varepsilon_{5s}$
S_{11}	Supply filter alert present	$A_{s,filter} = 1$
S_{12}	High Δp_r	$\Delta p_r > \varepsilon_{5r}$
S_{13}	Return filter alert present	$A_{r,filter} = 1$
S_{14}	Low Δp_s	$U_{s,fan} \geq 5$ and $\Delta p_s < \varepsilon_6$
S_{15}	Low Δp_r	$U_{r,fan} \geq 5$ and $\Delta p_r < \varepsilon_6$
S_{16}	Difference T_i and T_o	$\lvert T_o - T_i \rvert > \varepsilon_7$ or $T_i = NaN$ or $T_i < \varepsilon_{14}$ or $T_i > \varepsilon_{15}$
S_{17}	Difference T_p and T_i with ERW off	$(\lvert T_p - T_i \rvert > \varepsilon_7$ and $U_{erw} = 0\%)$ or $T_p = NaN$ or $T_p < \varepsilon_{14}$ or $T_p > \varepsilon_{15}$
S_{18}	Difference T_s and T_p with HCV & CCV closed	$(\lvert T_p - T_s \rvert > \varepsilon_7$ and $U_{hcv} = 0\%$ and $U_{ccv} = 0\%)$ or $T_s = NaN$ or $T_s < \varepsilon_{14}$ or $T_s > \varepsilon_{15}$
S_{19}	T_r outlier or missing data	$T_r = NaN$ or $T_r < \varepsilon_{14}$ or $T_r > \varepsilon_{15}$
S_{20}	T_e outlier or missing data	$T_e = NaN$ or $T_e < \varepsilon_{14}$ or $T_e > \varepsilon_{15}$
S_{21}	Difference T_r and T_e with ERW off	$\lvert T_r - T_e \rvert > \varepsilon_7$ and $U_{erw} = 0\%$
S_{22}	Comparison U_{hcv}, T_s and T_{set}	$(U_{hcv} > 0\%$ and $T_s > T_{set} + \varepsilon_8)$ or $(U_{hcv} < 100\%$ and $T_s < T_{set} - \varepsilon_8)$ or $(U_{hcv} > 0$ and $U_{erw} < 100)$ or $U_{hcv} < 0\%$ or $U_{hcv} > 100\%$
S_{23}	Difference T_s and T_{set}	$\lvert T_s - T_{set} \rvert > \varepsilon_9$
S_{24}	Heating coil pump error present	$A_{hc} = 1$
S_{25}	Low heating coil efficiency	$(T_s - T_p) < \varepsilon_{10} * \frac{U_{hcv}}{100}$ and $U_{hcv} > 0\%$
S_{26}	Comparison T_p and T_s with HCV closed	$U_{hcv} = 0\%$ and $T_s > T_p + \varepsilon_{11}$
S_{27}	Comparison U_{erw}, T_s and T_{set}	$(U_{erw} < 100\%$ and $T_s < T_{set} - \varepsilon_{12}$ and $T_r > T_o)$ or $(U_{erw} > 0\%$ and $T_r < T_o)$ or $(U_{erw} > 0\%$ and $T_s > T_{set} + \varepsilon_{12})$ or $(U_{hcv} > 0$ and $U_{erw} < 100)$ or $U_{erw} < 0\%$ or $U_{erw} > 100\%$
S_{28}	Low ERW efficiency	$U_{erw} > 0\%$ and $(\frac{T_r - T_e}{T_r - T_i} < \varepsilon_{13} * \frac{U_{erw}}{100})$
S_{29}	ERW alert present	$A_{erw} = 1$
S_{30}	p_s outlier or missing data	$p_s = NaN$ or $(p_s < \varepsilon_{16})$ or $(p_s > \varepsilon_{17})$
S_{31}	p_r outlier or missing data	$p_r = NaN$ or $(p_r < \varepsilon_{16})$ or $(p_r > \varepsilon_{17})$
S_{32}	Δp_s outlier or missing data	$\Delta p_s = NaN$ or $(\Delta p_s < \varepsilon_{16})$ or $(\Delta p_s > \varepsilon_{17})$
S_{33}	Δp_r outlier or missing data	$\Delta p_r = NaN$ or $(\Delta p_r < \varepsilon_{16})$ or $(\Delta p_r > \varepsilon_{17})$
S_{34}	High or low T_{set}	$T_{set} < \varepsilon_{18})$ or $T_{set} > \varepsilon_{19})$

Table 4.4: Symptoms detection rules of the DBN.

5
Experimental validation

In this Chapter, the developed DBN method is validated with fault experiments. First, the fault experiments that were conducted on the case study AHU are described. Then, the setup used for diagnosing faults for the experimental dataset is outlined. This is followed by the results of applying the developed DBN to the gathered data from the experiments. The chapter concludes with a discussion on the impact of the different diagnosis periods and DBN structure on the diagnosed faults.

5.1. Description of experiments

The developed DBN was validated with experiments conducted at the case study building in March 2024. The experiments were conducted during weekends when no occupants were inside the building, so the experiments could be performed without causing any disturbance.

Faults were introduced by changing controls in the BMS. For example, the first experiment changed the fan control signal to 30%. In this case, the DBN should diagnose an incorrect control signal fault. Since the faults were introduced via the BMS, no BMS alerts were present in the BMS on the days that experiments were conducted. Additionally, the experiments could not simulate ERW, HC and fan failure faults. This was not possible because the case study AHU was used for normal operation on weekdays.

The performed experiments are listed in Table A.1. In total, thirteen experiments were carried out on four different days. Control faults were simulated where the control signal was set either too low or too high, as well as incorrect setpoint faults and sensor bias faults. The experiments were limited to the supply side of the AHU, as the return side components were expected to show similar results.

In the rest of this section, the control signals and sensor values during each experiment are shown. This allows interpretation of the diagnosed faults presented in Section 5.3.

Figure 5.1 shows the first two experiments, which changed the supply fan control signal to 30% and 70% on March 23rd. As should be expected, the measured pressure changes because of the fan control signal. The fan does not supply the required pressure of the setpoint. During the first experiment, the measured supply pressure lies below the setpoint and during the second setpoint it lies above.

The ERW control signal was set to 0% during the experiment shown in Figure 5.2. Because of the incorrect ERW control signal, the HCV is opened to reach the desired supply temperature setpoint without the ERW. However, the HCV control signal is unstable, also leading to an unstable supply temperature. Normally, the HCV would not be opened while the ERW is turned off, which may have impacted the HCV control rule calculation. The control signal to the HCV can be found to correspond to a delayed increase or decrease in supply temperature in the figure.

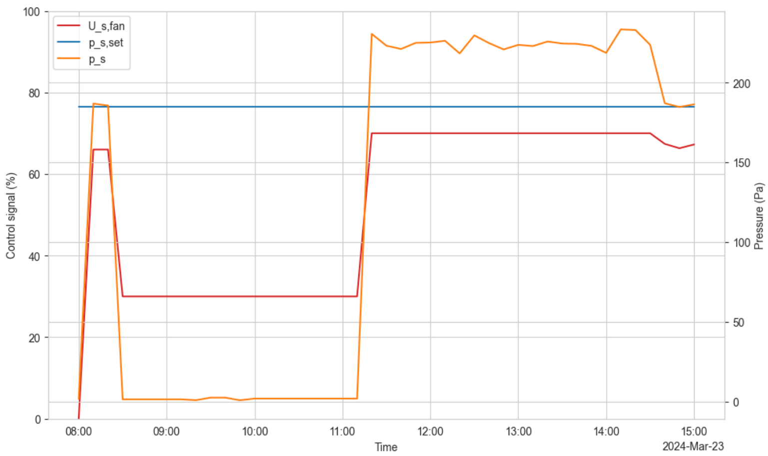


Figure 5.1: Supply fan control experiments conducted on March 23rd. The control signal of the supply fan is compared to the measured supply pressure and pressure setpoint.

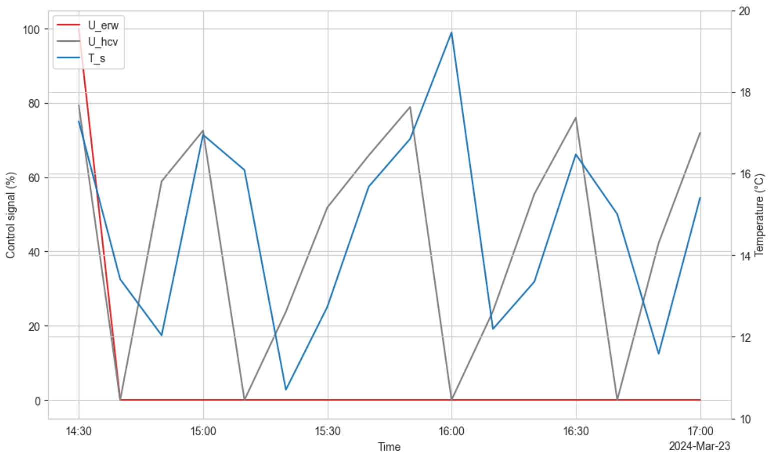


Figure 5.2: ERW experiment conducted on March 23rd. The control signal of the ERW is compared to the control signal of the HCV and the preheating and supply sensors.

The supply pressure setpoint was set to 235 Pa and 135 Pa instead of 185 Pa in two experiments on March 24th. The experiment is shown in Figure 5.3. The control signal to the supply fan is adjusted when the setpoint changes.

The pressure sensor at the supply side of the AHU was set to 130 Pa in the experiment shown in Figure 5.4 and to 200 Pa in the experiment shown in Figure 5.5. The fan control signals are correctly set to 100% and 0% by the AHU, aiming to increase or decrease the measured pressure by controlling the fan. The DBN does not include bias faults for the pressure sensors, however, this experiment was still conducted to find if different faults would be diagnosed instead.

In Figure 5.6, the control of the HCV during the experiments on the 29th of March is shown. The supply temperature data is also shown in the figure. As expected, the supply temperature drops when the heating coil valve is closed at 09:25. It then slowly increases after the control signal is set to 30% at 11:24. From 17:00 until 20:00 the control signal is set to 100%, which should increase the supply temperature substantially. Surprisingly, this is not the case. When the control signal is set by the BMS again at 20:00, the supply temperature suddenly sharply increases. Therefore it seems that the HCV was not actually opened to the intended control value.

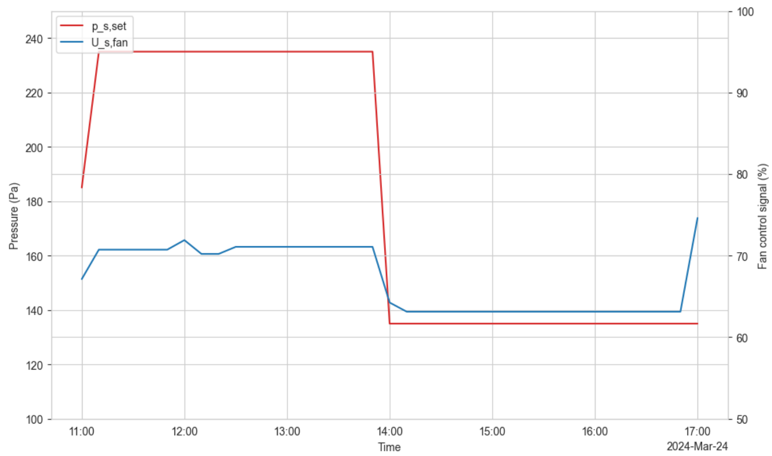


Figure 5.3: Supply pressure setpoint experiments conducted on March 24th. The supply pressure setpoint is compared to the control signal of the supply fan.

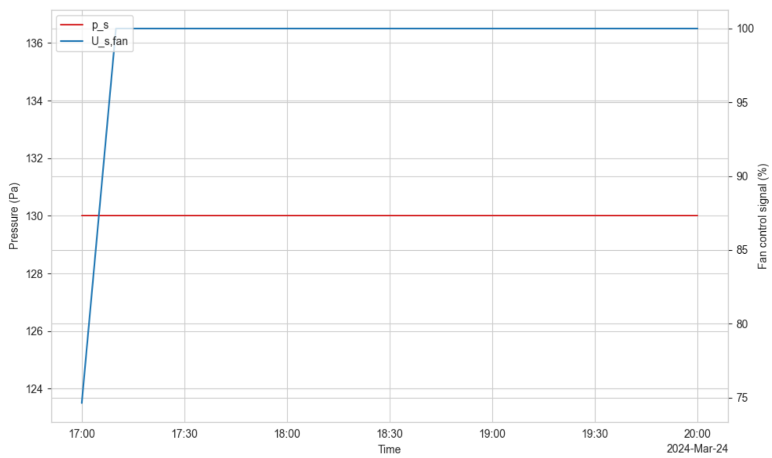


Figure 5.4: Supply pressure sensor experiment conducted on March 24th. The sensor value was changed to 130 Pa.

The temperature setpoint was set to 23 °C and 17 °C in two experiments on the 30th of March. These experiments are shown in Figure 5.7. During the first experiment, the HCV is increasingly opened until the high temperature setpoint is reached. The supply temperature then stabilises around the required temperature. When the temperature setpoint becomes 17 °C, the supply temperature does not stabilise. Instead, it fluctuates between 12 °C and 20 °C. As Figure 5.7 shows, this is because of the unstable HCV and ERW controls. The ERW is sometimes turned off, which is inefficient as the supply temperature can also be decreased by lowering the control signal to the HCV. Additionally, the HCV signal is set to 0% instead of being reduced slightly. This causes the supply temperature to drop around 5 °C multiple times.

In the final experiment, the measured value of the supply temperature sensor was set to 17 °C. The temperature value and control signals of the HCV and ERW are shown in Figure 5.8. The control signals correspond to expected behaviour. The HCV control signal is continuously increased to reach the desired setpoint temperature, but because the measured value has been set to 17 °C this is not achieved. The ERW is also turned on immediately after the start of the experiment.

The previous figures showed that for some experiments, the AHU did not respond to the altered control signals as expected. Therefore, the faults that are diagnosed should be different faults from the faults intentionally introduced during the experiments. For example, when the HCV control signal was set to 100%, the supply temperature remained below the setpoint temperature. Consequently, this should be diagnosed as a

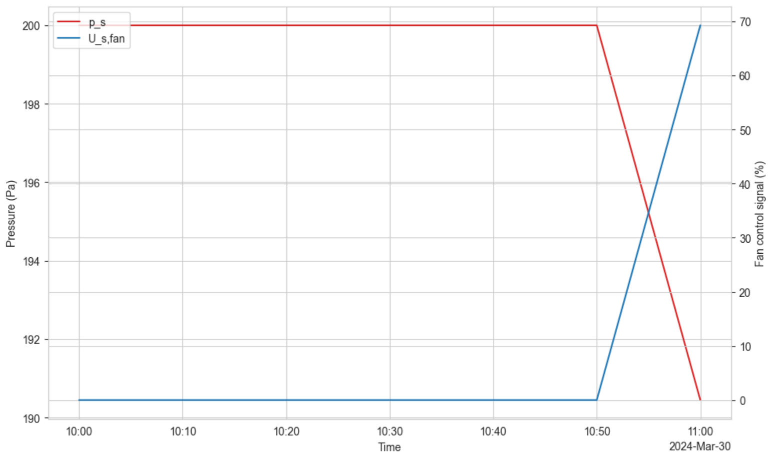


Figure 5.5: Supply pressure sensor experiment conducted on March 30th. The sensor value was changed to 200 Pa.

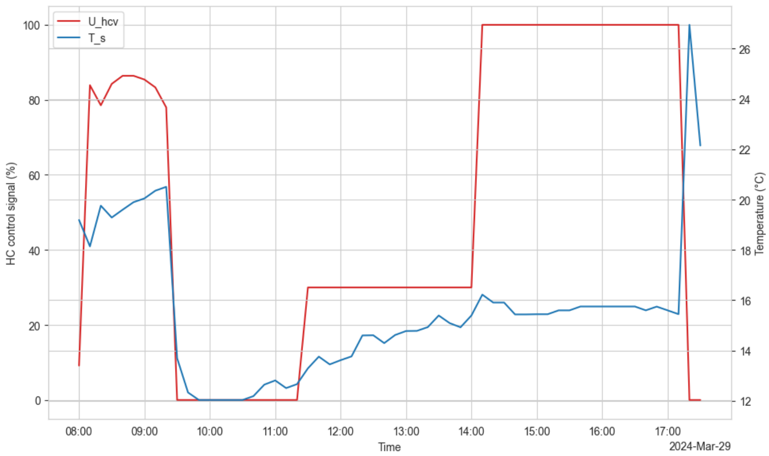


Figure 5.6: HCV control experiments conducted the 29th of March. The control signal was set to 0%, 30% and 100%.

HC failure fault instead of a HCV control fault. It is possible that a fail-safe mechanism in the BMS prevented the AHU from fully opening the HCV, which could explain why the supply temperature did not increase as expected. For the majority of experiments, however, the BMS adjusted the control signals as expected based on the faults that were introduced.

5.2. Diagnosis setup

The setup for the diagnosis of the data from the experiments is outlined in this section. First, the sensor data that was available is described, as well as how issues regarding missing data were handled. This is followed by the diagnosis periods and symptom detection thresholds that were used for performing the diagnosis.

Fortunately, the majority of sensor data from the experiments was available. In contrast to the historical data, the return temperature sensor data was now also stored in 10-minute intervals. The sensor data that was missing were the supply/return pressure and supply temperature setpoints, as well as outdoor temperature data.

The outdoor temperature data was replaced with weather data from the KNMI, the Dutch National Weather Service. The temperature was recorded at the Rotterdam The Hague Airport, located approximately 6.5 kilometres from the case study building, at one-hour intervals. Therefore, the data was interpolated to align with the required 10-minute sampling interval.

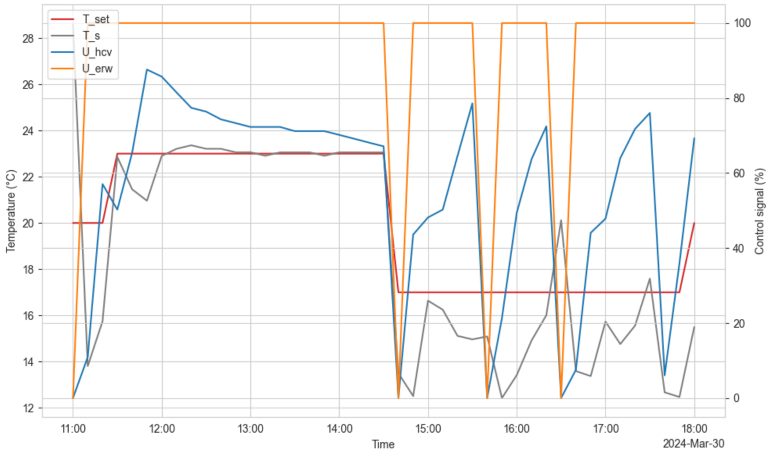


Figure 5.7: Two temperature setpoint experiments conducted on the 30th of March. The setpoint was set to 23 °C and 17 °C.

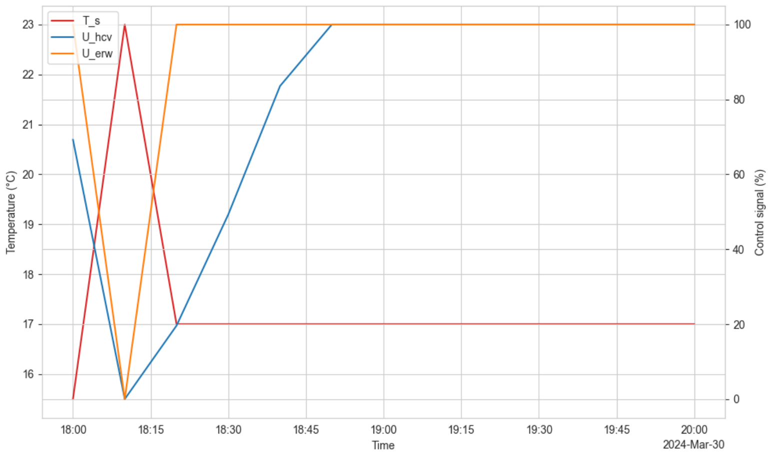


Figure 5.8: Supply temperature sensor experiment conducted on the 30th of March. The measured value was set to 17 °C.

In order to replace the setpoint data, values were added to the dataset manually. The temperature setpoint was replaced with a value of 20 °C, except during the temperature setpoint experiments on the 30th of March, when they were set to 23 °C and 17 °C. The return pressure setpoint was replaced with a 200 Pa value for all experiments. For the supply pressure setpoint 185 Pa was used for all experiments except for the supply pressure setpoint experiments on the 24th of March.

Different intervals for symptom detection are compared to establish which diagnosis period should be considered for real-time fault diagnosis of AHUs. The diagnosis is performed on 1-, 3-, 6-, 12- and 18-sample periods. As the samples are recorded every ten minutes, the diagnosis interval ranges from ten minutes to three hours. These periods were chosen because the experiments were conducted over a maximum period of three hours. Additionally, the experiments were conducted consecutively for practical reasons, so the diagnosis period could not be extended further.

For each experiment, the first sample that was diagnosed was the first ten-minute interval following the start time of the experiment. The final sample diagnosed is determined by either the diagnosis period or the last 10-minute sample before the experiment's end time, if the diagnosis period was longer than the duration of the experiment. For three experiments with a shorter duration, the symptoms detected are thus the same for some of the longer diagnosis periods. However, the results can still be different because of the different threshold values. These shorter experiments were the U_{hcv} = 0%, p_s = 200 Pa and T_s = 17 °C experiments.

The experiments were also diagnosed with different symptom thresholds for each diagnosis period to allow comparison of their effect on the diagnosed faults. Symptom detection thresholds were set for each period as follows. The threshold for the total number of symptoms is defined by $\lceil \frac{p}{6} \rceil$, for p number of samples. For the consecutive symptom threshold, the threshold was set to one sample for the 1-sample diagnosis period and to two samples for the other periods.

5.3. Diagnosed faults

Diagnosis period **Fault**	**10 minutes**	**30 minutes**	**1 hour**	**2 hours**	**3 hours**
$U_{s,fan}$ = 30%	$U_{s,fan}$, T_{er} bias, T_i, T_s	$U_{s,fan}$, T_{er} bias, T_i	$U_{s,fan}$, T_{er} bias, T_i, U_{hcv}	$U_{s,fan}$, T_{er} bias, T_i, U_{hcv}	$U_{s,fan}$, T_i, U_{hcv}
$U_{s,fan}$ = 70%	$U_{s,fan}$, U_{hcv}	$U_{s,fan}$, U_{hcv}	$U_{s,fan}$, T_i, U_{hcv}	$U_{s,fan}$, T_i	$U_{s,fan}$, T_i
U_{erw} = 0%	T_{er} bias, T_p	T_{er} bias, T_p	T_{er} bias, T_p	T_{er} bias, T_i, T_p	T_{er} bias, T_i, T_p
$p_{s,set}$ = 235 Pa	$p_{s,set}$	$p_{s,set}$	$p_{s,set}$	$p_{s,set}$, U_{hcv}	$p_{s,set}$
$p_{s,set}$ = 135 Pa	$U_{s,fan}$, $p_{s,set}$, T_i	$p_{s,set}$, T_i	$p_{s,set}$, T_i	$p_{s,set}$	$p_{s,set}$
p_s = 130 Pa	$U_{s,fan}$, $U_{r,fan}$, T_i, U_{hcv}	Damper$_s$, T_i	Damper$_s$, T_i	Damper$_s$, T_i	Damper$_s$, T_i
p_s = 200 Pa	$U_{s,fan}$, U_{hcv}	$U_{s,fan}$, U_{hcv}	$U_{s,fan}$, T_i, U_{hcv}	$U_{s,fan}$, T_i, U_{hcv}	$U_{s,fan}$, T_i, U_{hcv}
U_{hcv} = 0%	T_s, U_{hcv}	T_s, U_{hcv}	T_s, U_{hcv}	T_s, U_{hcv}	T_s, U_{hcv}
U_{hcv} = 30%	U_{hcv}	U_{hcv}	U_{hcv}	U_{hcv}	U_{hcv}
U_{hcv} = 100%	T_i, HC	T_i, HC	T_i, HC	T_i, HC	T_i, HC
T_{set} = 23 °C	T_{set}	T_{set}	T_{set}, U_{hcv}	T_{set}	T_{set}
T_{set} = 17 °C	T_{er} bias, T_p, T_{set}	T_{set}, U_{hcv}	T_{set}, U_{hcv}	T_{set}, U_{hcv}	T_i, T_{set}, U_{hcv}
T_s = 17 °C	T_i, U_{hcv}	T_i, U_{hcv}	T_i, U_{hcv}	T_i, U_{hcv}	T_i, HC

Table 5.1: Diagnosed faults for all of the experiments with five different diagnosis periods.

In this section, the results of applying the developed DBN with different diagnosis intervals to the data gathered from the experiments are presented. Table 5.1 shows the faults diagnosed with the DBN for each diagnosis period. The posterior probabilities of all faults for each experiment are shown in Figure 5.9.

Several faults are diagnosed in multiple experiments, unrelated to the fault that was introduced. The T_i fault is diagnosed often, which is not surprising as this fault was also present in the historical data. This is also the case for the U_{hcv} fault, which was also diagnosed in 30% of all historical days. The T_{er} bias faults also appear often in the diagnosed faults. This was not the case for the historical data, because this symptom could not be diagnosed due to the missing return temperature data and the consecutive symptom threshold. Symptom S21 is linked with a strong conditional probability to this fault. It is detected when there is a difference between the return and exhaust temperatures when the ERW is turned off, as the temperatures should be similar in that case. Therefore the number of samples where this symptom can be detected is limited to samples where the ERW is turned off. This explains why it is detected during the experiment where the ERW control signal is set to 0%.

For the following eight experiments, faults were diagnosed correctly in all diagnosis periods. In both fan control experiments, the incorrect control signal fault is correctly diagnosed. The high and low pressure setpoint faults are also detected as well as the HCV control signal faults in the 0% and 30% experiments. Finally, the temperature setpoint was diagnosed in both experiments in all diagnosis periods.

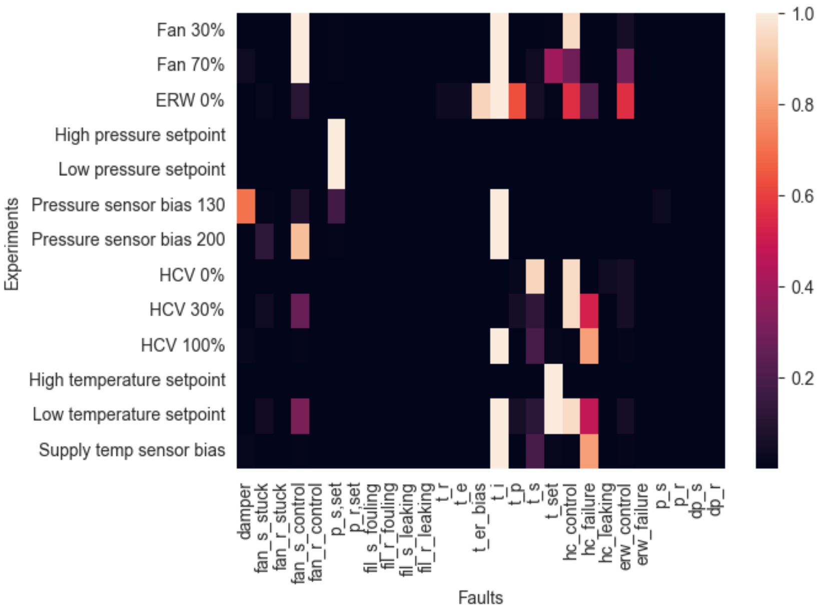


Figure 5.9: Inferred posterior fault probabilities of the original DBN for each experiment with a 3-hour diagnosis period.

Bias faults for the pressure sensors were not included in the DBN, only missing data or outliers. Therefore the frozen sensor values during the pressure sensor experiments could not have been diagnosed. However, the DBN should still recognise that the AHU is operating with faults, as the stuck pressure sensor value caused several symptoms to be detected. During the p_s = 130 Pa experiment, the closed damper fault is diagnosed in all diagnosis periods longer than 10 minutes. This is also a component fault, similar to the pressure fault which also causes a lower pressure and deviation from the setpoint. For the p_s = 200 Pa experiment, the supply fan control fault is diagnosed. This does not correspond to the control signal shown in Figure 5.5, which is correctly set to 0% based on the high measured pressure.

During the experiment where the heating coil control signal was set to 100%, the supply temperature did not increase as expected. This is shown in Figure 5.6. The fault that was detected for this experiment was the heating coil failure fault, instead of the HCV control fault. This aligns with the control signal, which could not be increased anymore to reach the desired supply temperature. Therefore, the DBN considers the HC to be failing as the temperature is not increasing as intended. This shows that the fault can be diagnosed even without a failure alert.

Two faults were not diagnosed correctly by the DBN. These are the ERW control and supply temperature sensor faults. During the ERW experiment, the control signal was set to 0%. This fault was not diagnosed with any of the detection periods. To determine what could have caused this, we can start by considering which symptoms have been detected for this experiment. Three symptoms related to the ERW control fault were detected in all diagnosis periods: symptoms S22, S23 and S27. Therefore the symptoms were detected properly related to this experiment, but the DBN did not diagnose the ERW control fault based on the detected symptoms.

The calculated posterior probabilities of the ERW control and HCV control faults are shown in Figure 5.9. Both posterior probabilities were 43.7%, which is below the 60% threshold used for considering faults. It seems that the DBN calculated both faults to be likely, but could not choose between the two faults because the three symptoms could be caused by either control fault. The reason that both faults were linked to symptoms S22 and S27 was the part comparing the HCV and ERW control signals. If the HCV is opened (U_{hcv} > 0%), the ERW should be running at maximum capacity (U_{erw} = 100%). If this is not the case, both control signals are incorrect and thus both the ERW control and HCV control fault should be diagnosed by the DBN.

To remedy this issue, the DBN structure was adapted. The following part of the symptom detection rules from symptoms S22 and S27 was removed and added to the new symptom S35: $U_{hcv} > 0\%$ and $U_{erw} < 100\%$. The structure of this revised DBN is shown in Figure B.1 and the corresponding symptom rules are listed in Table B.1. Both ERW and HCV control faults are linked with a strong conditional probability to symptom S35, as both controls are incorrect if the HCV is opened while the ERW is not running at maximum capacity.

Diagnosing the ERW experiment again with the revised DBN, resulted in the HCV and ERW control faults being diagnosed with all diagnosis periods. The other faults diagnosed were equal to the faults diagnosed with the original DBN.

These results have shown that symptom rules should not include the same parts, as the DBN will then not be able to infer a high posterior probability for related faults.

The second fault that was not diagnosed is the supply temperature sensor fault. Symptom S18 is linked strongly to this fault, which is detected when there is a large difference between the preheating and supply temperatures while the HCV is closed ($U_{hcv} = 0\%$). This symptom was not detected during the conducted experiment, because the HCV was open during all samples, as can be seen in Figure 5.8. Instead of the supply temperature fault, the HCV control or HC failure fault was diagnosed, depending on the diagnosis period. The reason for this is also shown in Figure 5.8. For the first three samples following the experiment start time, the control of the HCV is not set to the maximum value. Therefore, symptom S22 is detected as the HCV can be further opened to increase the supply temperature. From 18:50 onwards, the HCV control signal is set to 100%, while the supply temperature still does not meet the setpoint due to the introduced fault. This is recognised by the DBN as a HC failure fault only with the longest diagnosis period. For this period the samples considered are the same as for the 12-sample period, but the difference lies in the threshold used for the total number of symptoms. The shorter diagnosis periods diagnose the HCV control fault since two samples detect the symptom consecutively, at 18:30 and 18:40. For the 3-hour diagnosis period, the symptom is not considered detected since the threshold for the total number of samples is set to three. Therefore, the HC failure fault is diagnosed instead of the supply temperature fault, which does not have a strong link to either of these symptoms.

5.4. Impact of the diagnosis period

An additional objective of the experiments was to evaluate the diagnostic outcomes across different diagnosis periods. As illustrated in Table 5.1, there are some differences between the faults diagnosed across these periods. However, there are no instances where the introduced fault was partially identified; faults are either correctly identified in all periods or not identified in any. Remarkably, even the shortest diagnosis period of 10 minutes, consisting of just a single sample, is sufficient to accurately diagnose most of the introduced faults.

However, differences can be found when considering the diagnosis of other faults beyond the ones intentionally introduced during the experiments. Specifically, the 10-minute interval tends to yield diagnoses of faults that are not detected in longer intervals. This is the case for the $U_{s,fan} = 30\%$, $p_{s,set} = 135$ Pa, $p_s = 130$ Pa and $T_{set} = 17\ ^\circ$C experiments. These findings suggest that shorter diagnosis periods may be more sensitive to transient anomalies, resulting in more false positives. For longer periods this may not happen as often because of the symptom thresholds.

Another important consideration regarding the diagnosis period is that diagnosis was only performed on the first part of the experiment data matching the diagnosis period. For instance, with the 10-minute diagnosis period, only the first 10 minutes of data was diagnosed. An alternative approach would have been to diagnose every 10-minute sample during each experiment and then combine all of the diagnosed faults. This would likely result in additional faults being diagnosed, increasing the chance of additional false positives.

An appropriate diagnosis period for practical applications can be determined based on the presented results. A one-hour diagnosis period appears to be a reasonable compromise. This interval allows multiple diagnostic checks throughout the day, enabling rapid identification of any faults in the AHU. In particular control faults can be diagnosed in a short time frame, potentially leading to energy savings when they are swiftly addressed. Furthermore, applying the one-hour period still allows consideration of multiple samples per diagnosis, which helps to mitigate the risk of false positive diagnoses. Therefore, with this time span the accuracy and reliability of the diagnosis are maintained for practical usage.

5.5. Sensitivity analysis

A sensitivity analysis of the DBN structure is presented in this section. The DBN structure consists of conditional and prior probabilities which are used to infer posterior fault probabilities. By diagnosing faults using different probabilities, we can assess the impact of these parameters on the accuracy of the fault diagnosis. This can show how robust the DBN method is, as it should be able to diagnose faults correctly even when the probabilities are set sub-optimal.

If the outcomes are highly sensitive to the conditional and prior probabilities, it indicates that finding the correct probabilities for the DBN will require careful calibration and possibly extensive research. Conversely, if the diagnosed faults are not strongly influenced by the conditional and prior probabilities, it shows that the DBN method can generate results reliably. In this manner, the sensitive analysis will demonstrate the robustness of the diagnostic method.

The revised DBN with 35 symptoms is compared to three different DBN structures to determine the sensitivity. The three altered DBNs are based on the revised DBN, with equal symptom detection rules and links between symptoms and faults, only the prior or conditional probabilities were altered. Each DBN was altered as follows. The posterior probabilities of Figure 5.10b were diagnosed with a DBN where all conditional probabilities were set to weak probabilities of 30%. For Figure 5.10c, all conditional probabilities were changed to strong probabilities of 90%. Finally, for the DBN in Figure 5.10d, all prior probabilities of faults occurring were set to 5% instead of 2% for component faults.

The inferred prior probabilities of these four DBNs are shown in Figure 5.10. The results were generated by diagnosing the longest diagnosis period for each experiment.

The calculated posterior probabilities in Figure 5.10 show the effect of changing the conditional probabilities of the DBN. When the DBN consists of only weak links, faults that previously showed high posterior probabilities resulted in lower posterior probabilities. Conversely, faults that had low posterior probabilities in the revised DBN see an increase in their posterior probabilities. In contrast, when the DBN is constructed with only strong links, the opposite effect is observed: faults with high posterior probabilities become even higher, while those with low posterior probabilities decrease further. One of the exceptions is the damper fault, which is not diagnosed for either of the altered DBNs for the experiment with the supply pressure sensor set to 130 Pa. Therefore, this fault seems to be diagnosed because of the two different conditional probabilities in the original DBN.

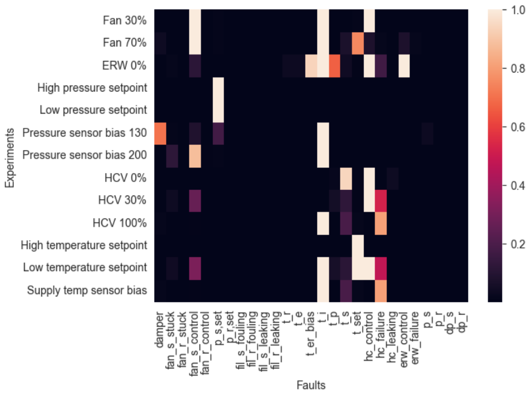
For the DBN with only weak links, false negatives are present for some experiments which were absent for the revised DBN. This is the case for the HC failure fault for two experiments and the pressure sensor bias at 130 Pa experiment. All control faults are still diagnosed correctly, however.

For the DBN with only strong links all control faults are also diagnosed correctly. The DBN structure was able to diagnose the pressure setpoint fault for the pressure sensor bias of 130 pascal, which is an improvement over the DBN with only weak links, which did not diagnose any faults. However, this DBN frequently diagnosed the HC failure fault alongside the HC control fault. In two experiments, only the HC control fault should be diagnosed, indicating that this DBN structure is more prone to diagnosing false positives.

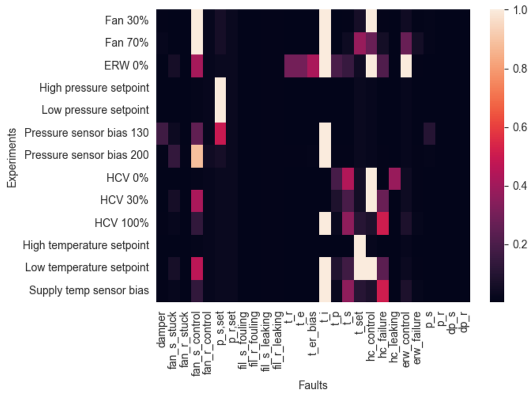
The impact of changing the prior probabilities to 5% is less significant compared to the effect of altering the conditional probabilities. All faults introduced during the experiments are still diagnosed, except for the 200

Pa pressure sensor bias experiment. However, the calculated posterior probabilities for the HC failure fault are higher for two of the experiments. Although the posteriors do not exceed the detection threshold and are therefore not considered diagnosed, they would have been diagnosed if the threshold had been set lower.

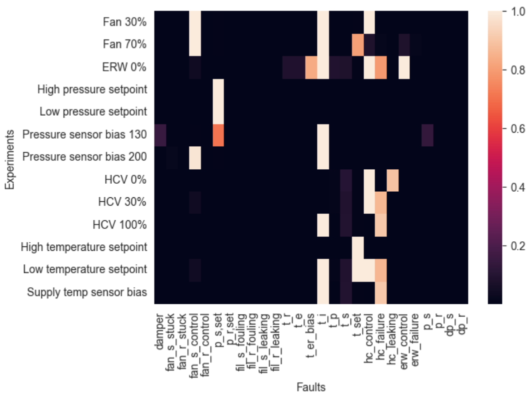
These findings suggest that differentiating the prior probabilities between control and component faults has a positive effect, as it helps to refine the accuracy of fault diagnosis by reducing false positives and improving the model's reliability in identifying the correct faults. The DBN with only strong links showed more false positives compared to the revised DBN and the DBN with only weak links had more false negatives. The results were less sensitive to the prior probabilities, as only one experiment was incorrectly diagnosed with the altered DBN.



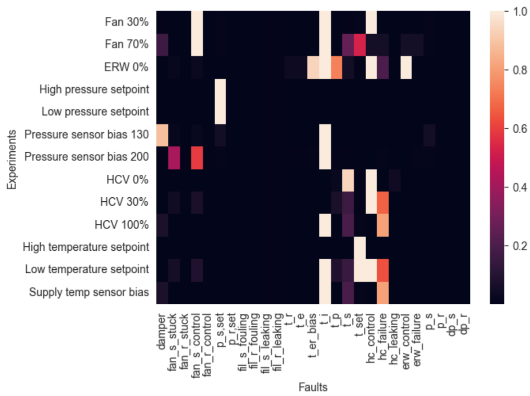
(a) DBN with added symptom S35.



(b) DBN containing only weak links.



(c) DBN containing only strong links.



(d) DBN containing only 5% prior probabilities.

Figure 5.10: Inferred posterior fault probabilities of different DBN structures, diagnosed for each experiment with a 3-hour diagnosis period.

6
Diagnosis of historical data

In this chapter, the diagnosis setup and results of analysing the historical case study data are shown. The chapter concludes with a discussion of the sensitivity of the results in regard to the symptom and fault thresholds.

6.1. Diagnosis setup

The setup for applying the developed diagnosis method to the available case study data from 2022 and 2023 is described in this section. This consisted of determining the diagnosis period, selecting the days that would be diagnosed and setting the symptom thresholds.

The diagnosis period was set to the maximum period, performing diagnosis once per day, to reduce processing time. This resulted in a period of 72 samples of 10-minutes, from 08:00 until 20:00, for each day.

Most of the days available in the 2022 and 2023 datasets were diagnosed, even though the DBN was developed to focus on the winter and in-between seasons. The method could also perform diagnosis during the summer season, however, because the cooling coil valve (CCV) control signal was taken into consideration for the symptom detection rules. It only would not be able to diagnose faults related to the CCV. According to operation data, the AHU was not operated during weekends and on several holidays. After filtering out these days, 506 days were left for applying the diagnosis.

The preprocessing of the data was performed as described in Section 4.3. The sensor data were not modified as part of the preprocessing process. The return temperature sensor data, which was shown in Section 3.2 to be available in 30-minute intervals in a portion of the dataset, was also chosen not be interpolated. This could have been done to align the data with the required 10-minute sampling rate. The reason for choosing not to interpolate the sensor data is that sensor data can also be missing in practice. Consequently, the missing sensor data is used to simulate cases of missing data in real-time applications of the DBN. If the decision was made to interpolate the data instead, additional symptoms could be detected that included this sensor, such as the ERW efficiency symptom (S28).

Finally, the daily symptom detection threshold was set to three samples and the consecutive symptom detection threshold to two samples. Therefore, each symptom was only considered if it was detected on two samples consecutively for more than three times per day. These values were chosen based on the diagnosis method of Chitkara, which considered faults to be diagnosed if they were detected three times consecutively.

6.2. Diagnosis results

In this section, the results of diagnosing the historical data with the previously described setup are presented. First, the detected symptoms are discussed. This is followed by a description of the diagnosed faults. The symptoms detected and faults detected on more than 10% of the days considered are marked in the DBN in Figure 6.1.

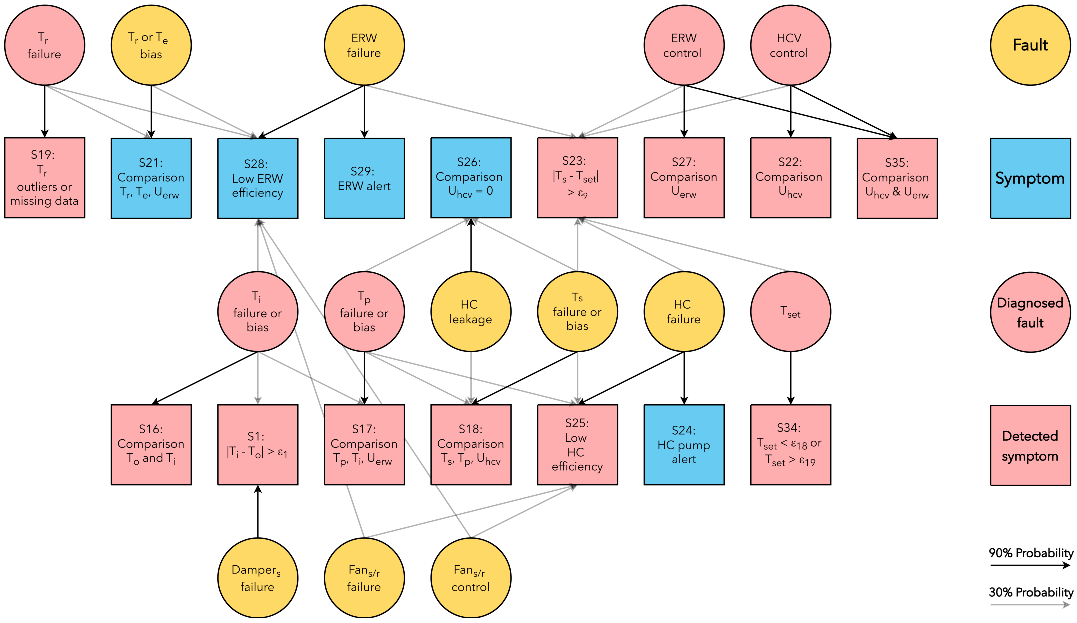


Figure 6.1: Symptoms detected and faults diagnosed for more than 10% of the samples for the historical data, shown in a part of the DBN.

6.2.1. Detected symptoms

In this section, the symptoms detected by the DBN are presented. Figure 6.2 shows in how many samples each symptom was detected, and for how many days they were considered for fault diagnosis, with the symptom thresholds that were used. There is a wide variance in the number of days the symptoms are detected; eleven symptoms were not detected for any days and two were detected for almost all days. Additionally, eleven symptoms were considered to be detected on more than 10% of the days, which are also shown in the DBN in Figure 6.1.

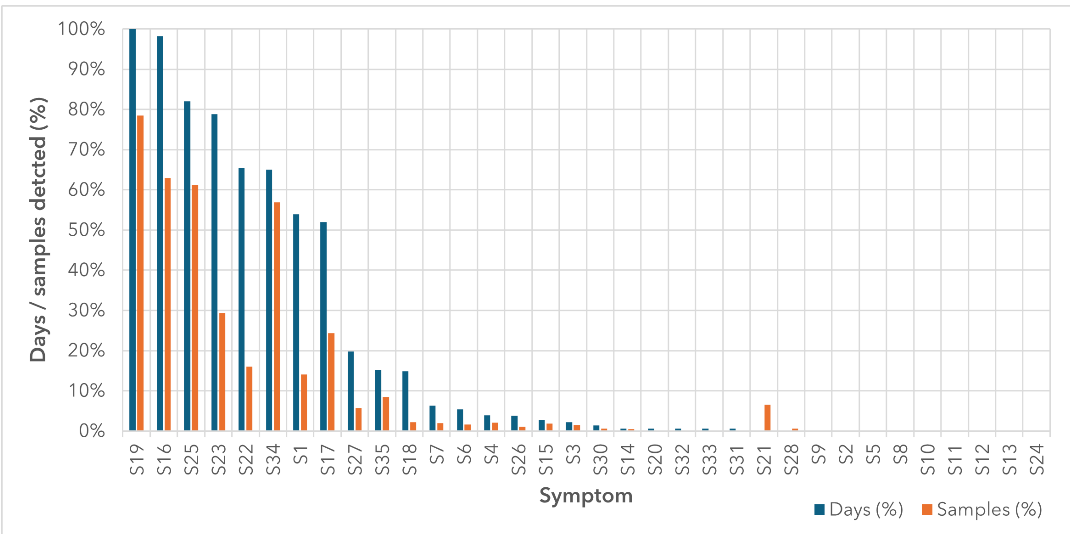


Figure 6.2: Symptoms detected for the historical dataset of 2022 and 2023.

The ratio between the samples and days shows how the symptom thresholds have affected symptom detection.

If the percentage of days detected is larger than the percentage of samples, the symptom is present on most days, but not many times per day. Therefore, increasing the total number of symptoms threshold could lead to these symptoms being detected less often. In contrast, if the percentage of samples detected is larger than the percentage of days, the symptom was detected on some days in the year, perhaps related to the season.

Interestingly, the ratio varied significantly per symptom. For example, symptoms S23 and S25 were both detected on a similar number of days, even though the number of samples was almost twice as large for S25 compared to S23.

The number of days each symptom was detected is presented in the remainder of this section, in order of detection rate. For each symptom, an explanation is provided on what may have caused the symptom to be detected.

Symptom S19 was detected for all days, indicating that return temperature data is missing or unrealistic. As previously shown in Figure 3.6, this sensor data was indeed missing or only available in 30-minute intervals. This results in the symptom being detected for two out of three samples for this last part, which is why it was accurately detected for all days considered.

Symptom S16 was detected if the inlet temperature sensor value is missing, unrealistic or biased compared to the outdoor temperature. As we can see from Figure 3.6, little data is missing for this sensor. The values are also not unrealistic, as shown in Figure 3.7. Therefore, the symptom seems to be detected by a difference of more than 1 °C between the outdoor and inlet temperatures.

Symptom S25 indicates that the supply temperature is lower than expected compared to the preheating temperature, based on the control signal to the HCV. A reason that this symptom was detected on 82% of the days could be that the supply or preheating temperature sensor is biased, or the heating coil is frequently unable to deliver the required temperature. It could also be caused by a delay between the HCV control signal and the measured supply air temperature.

Symptom S23 shows that the supply temperature is often more than 1 °C higher or lower than the setpoint temperature. This symptom was detected on 79% of all days and 29% of all samples. In the dataset, the supply temperature was less than 1 °C lower in 4923 samples and more than 1 °C higher in 5745 samples. This distribution is quite even, showing that the component controls are not biased to a higher or lower supply temperature compared to the setpoint. The symptom was likely detected because of incorrect HCV and ERW controls, as the symptoms indicating this (S22 and S27) are also detected frequently.

Symptom S22, indicating incorrect HCV control, was detected on 16% of the samples and 67% of the days considered. The low number of samples shows that the symptom was not detected on many samples per day. The symptom detection rule checks the signal for outliers; and compares the HCV control signal, supply temperature and temperature setpoint. From Figure 3.9, the control signal values have been shown not to include any outliers. Therefore, this symptom was detected because the HCV control should be increased or decreased to match the desired temperature setpoint.

Symptom S34 was also related to the temperature setpoint, however, this symptom was detected when the difference between the setpoint was not set between 19 °C and 22 °C. These temperatures were based on the control document of the AHU, which was written by the building operators. The symptom was detected often due to the setpoint being set lower than 19 °C, as can also be seen in Figure 3.7.

Symptom S1 was detected on 54% of the days considered. Similar to symptom S16, this symptom is detected when the difference between the outdoor and inlet temperatures is greater than a threshold. However, for this symptom, the threshold is 2 °C instead of 1 °C. The number of samples detected was small compared to the number of days, indicating that this symptom might be detected mostly during specific periods of the day, such as during the startup of the AHU.

Symptom S17 was detected on 52% of the days in the dataset. Since there was no missing data for the preheating sensor, this symptom was detected because of a biased inlet or preheating temperature sensor. As symptom S16 was detected more often, the inlet sensor was more likely biased.

Incorrect ERW control was detected by symptom S27 in 6% of the samples and 20% of the days considered. The symptom detection rule of symptom S27 contains four different rules, that are detected if: the signal contains outliers; the ERW should be turned on to meet the supply temperature setpoint; the ERW should be turned off to meet the supply temperature setpoint; the ERW should be turned off because the return temperature is lower than the outdoor temperature. As the control signal did not contain any outliers, shown in Figure 3.9, this symptom was detected because the control signal did not correspond to the measured temperatures and temperature setpoint.

Symptom S35 was added in the revised DBN, presented in Figure B.1. This symptom detection rule compares the control signals of the ERW and HCV, if the HCV is opened while the ERW is turned off: $U_{hcv} > 0$ and $U_{erw} < 100$. This symptom was detected on 8% of the samples and 15% of the days considered.

Symptom S18 is related to the supply sensor and was detected on 2% of the samples and 15% of the days. The ratio between the days and samples that this symptom was detected was the highest of all symptoms. This symptom can be detected due to missing data, outliers or a difference between the preheating and supply temperatures. Based on Figures 3.7 and 3.6, for this sensor few data was missing and no outliers were present. Therefore, it was detected because of a bias between the preheating and supply sensors when the heating and cooling coil valves were closed.

Four pressure-related symptoms were detected on a small percentage of days. Symptom S7 detected a difference of at least 5 Pa between the return pressure and setpoint for 6% of the days considered. Symptom S6 is similar to symptom S7, however, this symptom is only detected if it is possible to increase or decrease the control signal of the return fan to match the setpoint. Since symptom S7 was detected, the return pressure sometimes deviates from the setpoint and in these cases the control value should be adjusted, causing this symptom to be detected for 5% of the days considered.

Symptoms S3 and S4 are equal to symptoms S6 and S7, but for the supply side of the AHU instead of the return side. They are detected on 2% and 4% of the days considered, respectively. Notably, the differences between symptoms S4 and S7 and symptoms S3 and S7 are small, which shows that the difference between the pressure setpoint and the measured pressure was usually caused by incorrect control signals.

For the remaining symptoms, the low number of days that they are detected aligns with the historical data, as no alerts were known to be present during the diagnosed period. Additionally, the AHU was expected to operate fault-free.

No faults were expected to be present for the ERW in the AHU as well, which aligns with the low number of samples in which symptom S28 was detected, which indicates low ERW efficiency. However, that this symptom was not detected for any day at all is odd considering that it was detected for some samples. This was caused by the missing return temperature data, which made the model unable to detect the symptoms in two samples consecutively. Using the consecutive symptom threshold with missing sensor data thus causes symptoms where the sensor values are used to be unable to be detected. This should be considered as a consequence of using this threshold in practice.

6.2.2. Diagnosed faults

In this section, the faults that were diagnosed for the historical data are presented. Additionally, it elaborates on the correlation between the number of days the faults were diagnosed and the corresponding symptoms that were detected. Finally, the average posterior probabilities of the faults are discussed.

For each fault, the percentage of days for which it was diagnosed and the average inferred posterior probability

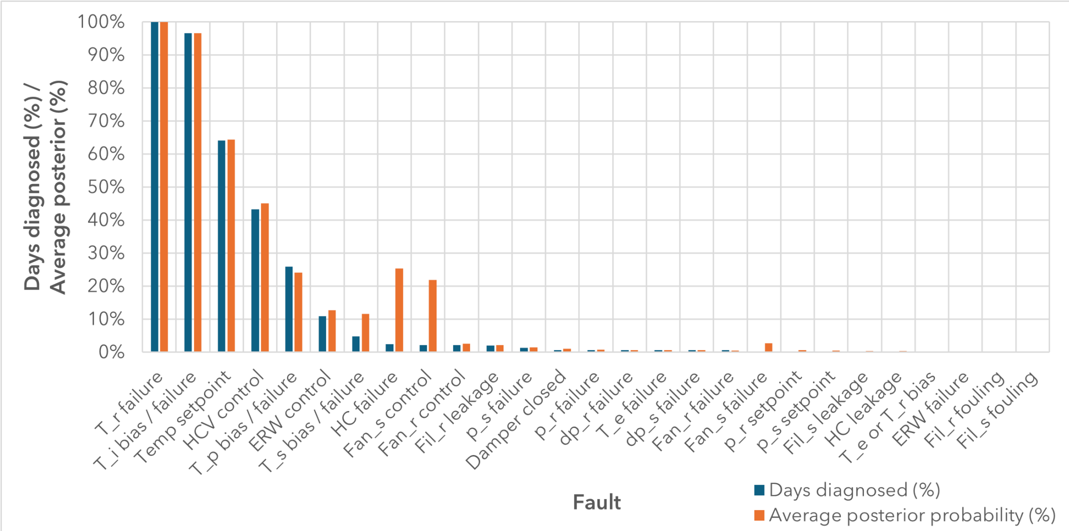


Figure 6.3: Faults diagnosed in the historical dataset of 2022 and 2023.

are shown in Figure 6.3. The faults detected on more than 10% of the samples are marked in the DBN in Figure 6.1. As can be seen in Figure 6.3, six faults were diagnosed in more than 10% of the days. In Chapter 8, the implications of the faults that were diagnosed are further discussed.

The fault most often diagnosed was the failing return temperature fault, diagnosed on all days in the dataset. This was expected with the missing data for this sensor, as explained in the previous section.

The inlet temperature sensor bias or failure fault was also diagnosed often, in 97% of the days in the dataset. This aligns with the large number of samples in which symptom S16, linked strongly to this fault, was detected. As mentioned in the previous section, symptom S16 was detected frequently because of a difference between the outdoor and inlet temperatures. Since the outdoor temperature came from on an online weather source included in the BMS, the inlet temperature sensor was more likely to be biased. However, the threshold used of 1 °C could also have been too low. This will be further discussed in Chapter 8.

The temperature setpoint fault was detected on 64% of the days, equal to symptom S34, linked strongly to this fault. This fault was diagnosed often due to the setpoint being set lower than 19 °C.

The HCV control fault and the preheating temperature bias or failure fault were detected on 43% and 26% of the days considered, respectively. The symptoms strongly linked to these faults (S22 and S17) were also detected frequently, 65% and 52% respectively. The lower percentages for the faults compared to the symptoms were likely caused by the presence or absence of other symptoms linked to the faults. Symptom S35 is also strongly linked to the HCV control fault and detected on fewer (15%) days. This could explain why the fault was only diagnosed on 43% of the days, as the DBN may infer that the fault is not present if only one of the two strongly linked symptoms has been detected. For the preheating sensor fault, symptom S17 is also connected with a weak link to the inlet sensor fault. As this fault was often diagnosed, the DBN seems to have inferred that symptom S17 was likely caused by the inlet sensor instead of the preheating temperature sensor.

The last frequently diagnosed fault is the ERW control fault, which was diagnosed for 11% of the days. The symptom linked strongly to this fault (S27), was diagnosed more frequently as well, for 20% of the days. This difference can be explained in the same manner as for the HCV control fault: by the lower number of days in which symptom S35 was detected.

The remaining faults were diagnosed infrequently, which aligns with the symptoms that were also detected infrequently. The damper fault is the only exception, as the strongly linked symptom S1 was detected on 54% of the days. However, for this fault, symptoms S4 and S14 are detected infrequently, which the fault is also linked to. Therefore, the inlet temperature fault is diagnosed correctly instead.

The average posterior values for the six faults diagnosed most often closely resemble the percentages of days they are diagnosed. This indicates that the posterior values per day were likely either close to 0% or 100%, which would result in an average value that resembles the percentage of days diagnosed.

For the three faults following these first six faults, the average posterior probabilities are relatively high, while the percentages of days diagnosed are only 2% to 5%. This is especially so for the HC failure fault, which was detected on 2% of the days with an average posterior probability of 25%. The relatively high posterior probability shows that the calculated fault probability was often higher than 0%, although insufficient to pass the fault probability threshold of 60%.

However, this fault being diagnosed infrequently aligns with the symptoms that were detected. No HC pump alerts were present, and the other two symptoms related to the HC failure fault S23 and S25 can also be caused by two other faults. For these two faults, the preheating temperature sensor fault and the HCV control fault, more related symptoms are detected frequently. These are symptoms S17 and S18 for the preheating sensor fault and symptoms S22 and S27 for the HCV control fault. Therefore, the fault is correctly diagnosed infrequently, as these faults should be diagnosed instead.

For the supply temperature and supply fan control faults, the high average posterior probability can also be explained by the detected symptoms. For the supply temperature fault, the preheating temperature sensor is diagnosed instead, which is connected to the same symptoms. For the supply fan control fault, only one symptom (S25) is connected that was detected for a larger number of samples. Since the fault should not be diagnosed only because symptom S25 was detected, the low number of diagnosed samples for the fault is correct. Therefore, for these faults, the higher posterior probability aligns with the expected diagnosis result.

6.3. Sensitivity analysis

In this section, a sensitivity analysis is presented of the results with respect to the total and consecutive symptom thresholds, as well as the fault thresholds. The thresholds were set to smaller and larger values to assess their impact on the detection of symptoms and diagnosis of faults. Faults that are diagnosed differently per configuration should be considered less certain, compared to faults that are detected on all configurations.

This approach also evaluates the robustness of the diagnosis method. In practice, the fault diagnosis method should be reliable enough to diagnose similar faults, even with somewhat different threshold values. Otherwise, the diagnosis would need to be performed multiple times, while the engineer performing the diagnosis would need to interpret the differences in results.

Additionally, this sensitivity analysis provides insight into the impact of the symptom thresholds compared to the fault thresholds.

The diagnosis was performed with two different configurations for the symptom thresholds. For the first configuration, both the total number of symptoms threshold and the consecutive threshold were set to one sample, effectively resulting in a diagnosis without any effect of the thresholds. The second configuration had the thresholds set to larger values, with the total number of symptoms set to six and the consecutive threshold set to three samples.

Because the total and consecutive symptom thresholds determine which symptoms are considered for fault diagnosis, they affect both the detected symptoms and consequently the diagnosed faults. In contrast, the fault thresholds only affect the total number of faults that were diagnosed, thus they do not affect the detected symptoms.

The number of days that symptoms are detected with the different symptom thresholds, shown in Figure 6.4, varies significantly per symptom. For some symptoms, such as symptoms S16 and S35, the difference between

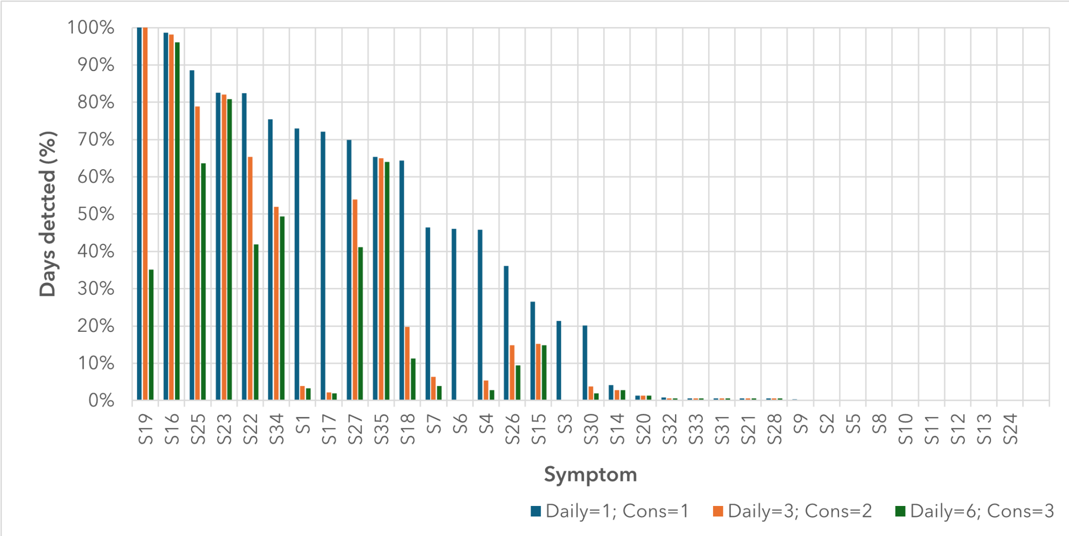


Figure 6.4: Symptoms detected for the historical data with different symptom threshold values.

the three configurations was minimal. For others, however, the difference between the configurations was extensive.

For eight symptoms, the number of days that the symptom was detected was reduced by more than half by using the original thresholds (with the total threshold set to two and consecutive threshold set to three), compared to not using the thresholds. The difference is most striking for symptoms S1 and S17, which detect biases for the inlet and preheating temperature sensor respectively. For symptoms S17 and S18, the effect of the consecutive threshold is expected to have an impact on the number of days they are detected, as these symptoms are only detected when the control signals of the ERW or HCV are 0%. Therefore, the symptoms can only be detected with the consecutive threshold if these control signals are 0% in multiple samples consecutively. This reduces the number of samples where these symptoms can be detected.

Symptoms S1 and S16 do not include any control signal in their symptom detection rules, which could be why S16 was detected on a similar number of days for all three configurations. However, this was not the case for symptom S1. This could be caused by the symptom only being present at the start-up of the AHU.

Other symptoms that are only detected often without the thresholds are symptoms S3, S4, S6 and S7. These symptoms indicate that the fan control signals should be altered to match the pressure setpoints; and that there was a difference of more than 5 Pa between the measured and setpoint pressure. The consecutive threshold could be causing these symptoms not to be detected because the building management system (BMS) correctly adjusts the control signals for the next sample if the setpoint was not being met. In this way, the threshold can help mitigate the delay between setpoint deviation and response by the BMS.

Comparing the original thresholds and the total and consecutive thresholds set to six and three, the difference was less significant. Four symptoms are detected on at least 10% fewer days with the adjusted thresholds: symptoms S19, S25, S22 and S27.

The difference was the largest for symptom S19, which detects missing return temperature sensor data. This was caused by the consecutive threshold that was set to three instead of two. For a large part of the dataset, the temperature sensor data was only available in 30-minute intervals, which caused this symptom to be detected for every two out of three samples. Therefore, with the consecutive threshold set to three, the symptom would not have been considered detected. This is one of the considerations that can be used to determine which value to use for the consecutive threshold: how frequently should a sensor fail consecutively before it can be considered to be malfunctioning?

The rules of the other three symptoms include control signals of the HCV and ERW. Therefore, these symptoms could also be detected less frequently because of the transient nature of the control signals and temperature data.

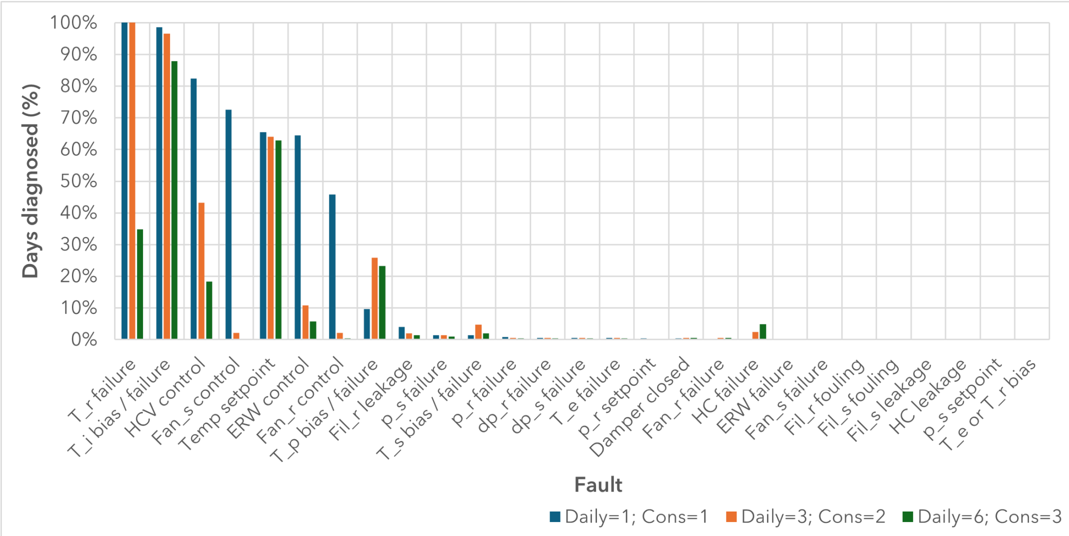


Figure 6.5: Faults diagnosed for the historical data with different symptom threshold values.

The faults diagnosed with the different total and consecutive thresholds are shown in Figure 6.5. Similar to the symptoms diagnosed, the difference between no thresholds and the original thresholds was the most significant. The faults diagnosed align with the detected symptoms shown in Figure 6.4: the control faults are detected more often without the thresholds, as well as the return temperature sensor failure fault.

Some faults are diagnosed more often with the thresholds, such as the preheating temperature sensor failure or bias and HC failure faults. This seems counterintuitive, however, this can be explained by the difference in symptoms that are detected. Symptom S25, indicating low HC efficiency, was frequently detected with all three configurations. This symptom is linked to both the preheating temperature sensor fault and the HC failure faults.

In contrast, the symptom indicating incorrect supply fan control (S3) was detected frequently without the thresholds. With the thresholds, however, S3 was not detected on any day. Consequently, the DBN diagnoses incorrect supply fan control in the absence of thresholds, while with thresholds, it identifies HC failures and preheating temperature sensor faults.

The return temperature sensor and HCV control faults are the only faults that were diagnosed on significantly fewer days with the largest thresholds. For the return temperature sensor fault, this aligns with the missing data for this sensor, as explained previously. However, for the HCV control fault, the significant difference suggests that this diagnosis may be largely influenced by the transitory nature of the data.

In conclusion, based on these results the symptom thresholds have been shown to reduce the number of control faults. This was likely caused by the consecutive threshold, requiring controls to be set incorrectly for multiple samples consecutively. Additionally, the value for the consecutive threshold can be determined by considering the number of samples that a sensor value can be missing before the sensor is considered to be failing.

The historical data was also diagnosed with two different posterior fault probability thresholds, of 50% and 70%, and compared to the original results with the 60% threshold. These thresholds determine at which calculated posterior probability a fault is considered to be diagnosed. If this threshold is set too low, faults can be diagnosed too often, resulting in false positives. However, if the value is set too high, the risk of false negatives

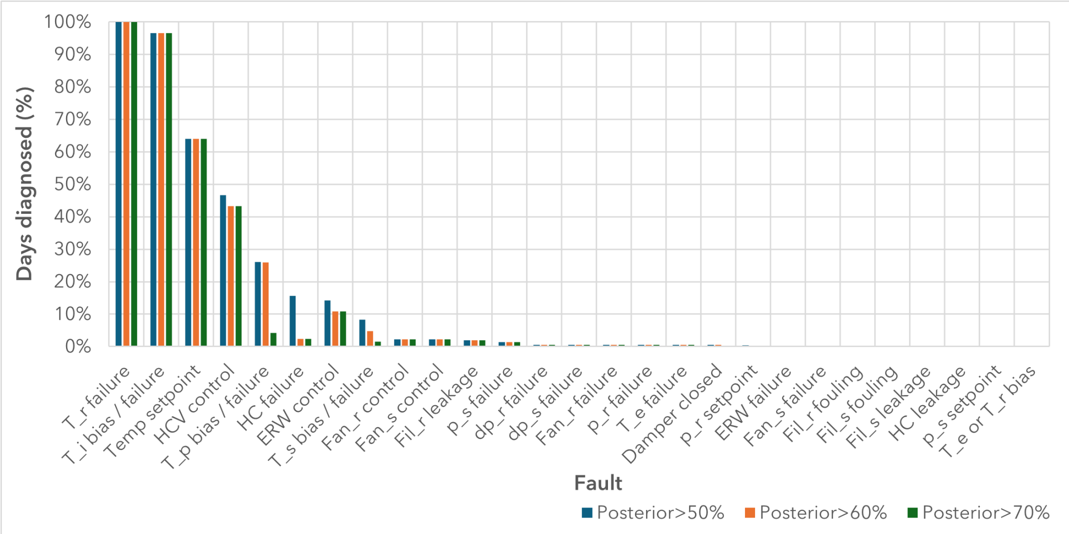


Figure 6.6: Faults diagnosed for the historical data with different posterior fault threshold values.

is increased.

The faults diagnosed with the three different threshold values are shown in Figure 6.6. The effect on the number of days each fault is diagnosed is minimal. For only two faults the threshold has a significant effect on the number of days diagnosed, namely for the preheating and heating coil temperature faults. This shows that the DBN was less confident that these faults caused the symptoms that were detected.

Comparing the results of the different posterior thresholds to the previous results with various symptom thresholds, the results were significantly more impacted by the symptom thresholds compared to the fault thresholds.

7

Real-time implementation

In this chapter the real-time fault diagnosis implementation in Python and the results of diagnosing real-time data are described. First, relevant background knowledge is provided of the libraries and frameworks that have been used. This is followed by the implementation of the diagnosis method and data collection. Finally, the results of diagnosing the collected data are described.

7.1. Description of Python frameworks

In this section, the Python frameworks used to perform real-time fault diagnosis are described.

In this thesis, Python is used for acquiring and preprocessing data, detecting symptoms and consequently detecting faults with a DBN. Preprocessing the data is done with the pandas library [34], which can import sensor data CSV files as DataFrame structures. The library can also be used to index and access the data by timestamp.

There are several Python libraries available for implementing Bayesian networks in Python. For this thesis, the library pgmpy was used [19]. In pgmpy, fault nodes can be created with prior probabilities as input. The symptom nodes need to be defined with the CPTs, which can be constructed based on the leaky-or method explained in Section 2.2.2. The library also includes functionality for probability inference through variable elimination. This can be used for computing posterior fault probabilities.

The acquisition of real-time sensor data is also implemented in Python. Data from the case study is stored in an Apache Kafka environment and accessed through the Python API provided by Confluent [10]. Apache Kafka is a distributed data streaming solution that operates with producers and consumers to generate and retrieve data independently [53]. The architecture of Kafka is shown in Figure 7.1.

Kafka is run as a cluster, consisting of multiple Kafka brokers that handle data storage and processing. The data is stored in the form of message logs in separate categories known as topics. The messages are also referred to as events. The topics are append-only immutable event logs, divided over multiple partitions for safety. The partitions are also divided over different brokers, allowing scalability. Each message consists of a timestamp, key, value and optional header parts. Within each partition, each message is also given a unique offset integer.

Producers and consumers are clients that can send or retrieve data to or from the topic. For producers, this is referred to as publishing messages to a topic. Consumers can first subscribe to a topic and can then consume from a topic to retrieve data. They use the offsets to keep track of which messages have been processed. Multiple consumers can be part of a consumer group, which divides the partitions among the consumers in the group. This enables one user to retrieve messages in parallel with multiple clients, dividing the workload. The messages are stored on the servers for a configurable period, allowing consumers to choose from which timestamp they want to consume. They can also retrieve all messages stored in a topic if desired. Finally, Kafka uses a centralised service called ZooKeeper to coordinate the Kafka brokers and store configuration settings.

The gathered data is stored locally for performing fault diagnosis. InfluxDB is used for this purpose, a database optimised for storing time-based data [39]. It has been specifically designed for storing measurements with timestamps, such as sensor information. Although the database can be deployed in the cloud, for this thesis InfluxDB OSS version 2 was used, which can be deployed locally [24]. This version is also open-source. InfluxDB can be interacted with through the InfluxDB UI, the InfluxDB command line interface (CLI) or the HTTP API. A Python client library is available that makes use of the HTTP API [16].

When InfluxDB is started for the first time, the initial setup process is initiated. This consists of choosing an organisation name, a bucket name and admin authorisation credentials. Afterwards, an API key can be created for interacting with InfluxDB through the Python client. The Python client library can write point objects to InfluxDB, consisting of a measurement name, tags and a field key and value pair. The points can be queried from InfluxDB by passing a Flux query as a string to the query API. Flux is the query language used by InfluxDB version 2.

In InfluxDB single records of data are referred to as points [23]. The points are organised in buckets and measurements. The buckets are named containers used for storing the measurements and the measurements are names used for grouping the points within the bucket. All points with the same measurement name should have the same tags, which are used for identifying points. Besides tags, a point also consists of a field key, field value and a timestamp. The field key is a string used to identify the type of value in the field value column, such as temperature or pressure. The field value can be either an integer, float, string or boolean. A collection of points for the same measurement and tags is called a series. The fields in InfluxDB are not indexed, therefore querying by field values is less efficient compared to querying by tag. Although adding tags is optional, adding them is beneficial for improving performance.

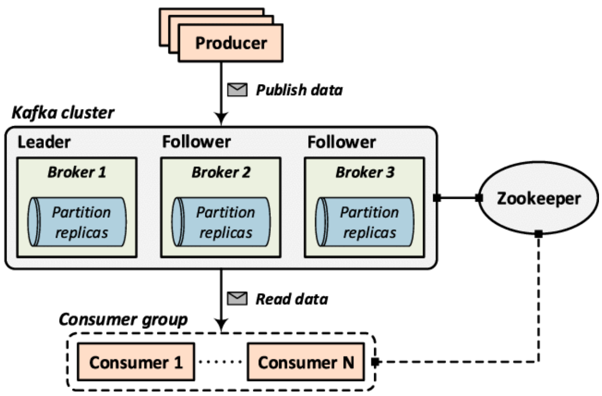


Figure 7.1: Overview of the Apache Kafka architecture by Javed et al [25].

7.2. Python implementation

In this section, the Python implementation for performing real-time fault diagnosis is described. The versions of all of the Python packages used in this thesis are listed in Appendix C.

The DBN was developed in Python, making use of the pandas and pgmpy packages for data preprocessing, detecting symptoms and inferring the posterior fault probabilities [19, 34]. The pgymp library was used for creating the Bayesian network, with prior probabilities added to fault nodes and conditional probability tables (CPTs) added to symptom nodes. The CPTs are constructed with the leaky-or model, explained in Section 2.2.2, based on the conditional probability parameters shown in Figure 4.1. The remainder of this section describes the real-time data collection process.

The data collection process for the case study AHU consisted of two steps. First, the data needs to be collected from the servers and then it has to be stored locally so fault diagnosis can be performed. The data related to the case study AHU was stored in an Apache Kafka environment.

For the case study, sensor data from various buildings was stored under one topic. The data could be retrieved through the Python Kafka API by building a consumer and consuming from this topic. The SASL (Simple Authentication Security Layer) framework was used for user authentication with a username and password encrypted with TLS/SSL. This security information is passed as a dictionary to the consumer during construction, with the following key/value pairs:

- 'sasl.mechanisms': 'PLAIN' - basic authentication with a username and password
- 'security.protocol': 'SASL_SSL' - the encryption protocol used
- 'sasl.username': the username of the user
- 'sasl.password': the password of the user
- 'group.id': the user's 'group.id', used to share the current offset between consumers created by the same user

The host and port of the Kafka broker are provided following the 'bootstrap.servers' key. Additionally, two parameters were added to ensure that messages were retrieved from the beginning of the datastream: 'enable.auto.commit': 'false' and 'auto.offset.reset': 'earliest'. By turning 'enable.auto.commit' off, the offset is only updated manually. This was mainly done for debugging purposes, so messages could be retrieved multiple times. The 'Auto.offset.reset' setting determines the position from which to consume in the message queue if no offset has been stored in the broker for the consumer. As we are interested in all data that is available, this position is set to the beginning of the partition.

With access, data can be retrieved in the form of messages by consuming from the topic. The messages consist of a timestamp, key and value. The timestamps and keys were not needed, as the sensor data timestamp and relevant meta-data are stored in the value part. The sensor data is stored in the value part serialised with the Avro schema. This schema is used instead of JSON, as it is more space-efficient [54]. The Kafka cluster contains schemas stored per topic, that can be used to convert the messages to JSON format. To access this schema from the Python client, three configuration parameters need to be provided: a URL in which the schema registry is stored, a public API key and a private secret key.

When the messages have been deserialised, the sensor data is available in JSON format. The sensor data is sent with Brick schema meta-data, which is a uniform open-source schema for representing metadata in buildings [3]. An example sensor data message is shown in Appendix D, for the case study AHU in building 28. The sensor data contains a building number; a timestamp in millisecond precision; and a list of equipment with sensor data. Each piece of equipment in the list contains a list of measurements, containing a sensor name and measured value related to the piece of equipment at the timestamp. In the example listing, the piece of equipment is the case study AHU and the measurement is a pressure sensor located in the AHU. For performing the fault diagnosis, the only data needed were the 'building number', 'timestamp', 'measurement_id' and 'value' tags.

InfluxDB was used for storing the sensor data retrieved from the Kafka cluster locally. The API token, organisation name and the localhost URL generated in the initial setup process were used for creating the InfluxDBClient in Python. This client allows interaction with InfluxDB through a 'write_api' and 'query_api'.

the Kafka messages cannot be written directly to InfluxDB, they first have to be converted to the format used by InfluxDB. The format used was a list of dictionaries representing points. Separate dictionaries were created for all sensor measurements by looping through all the messages and measurements, as shown in Listing 7.1. The dictionary contains the keys 'measurement', 'tags', 'fields' and 'time'. All created points use the same 'measurement' name. Two tags were added to identify sensor data: 'building' and 'sensor_name'. The 'measurements' value and 'timestamp' value in the message were added to the 'fields' value and 'time' value in the dictionary. The 'building and 'measurement_id_name' in the message were mapped as 'tags' values, with 'building' and 'measurement_name' as keys. The created points are passed to the Python InfluxDBClient 'write_api' method along with the bucket name, organisation name, access token and writing precision. The writing precision was set to milliseconds, which was also used for the timestamps in the messages. The resulting points can be viewed in the InfluxDB UI, shown in Figure 7.2.

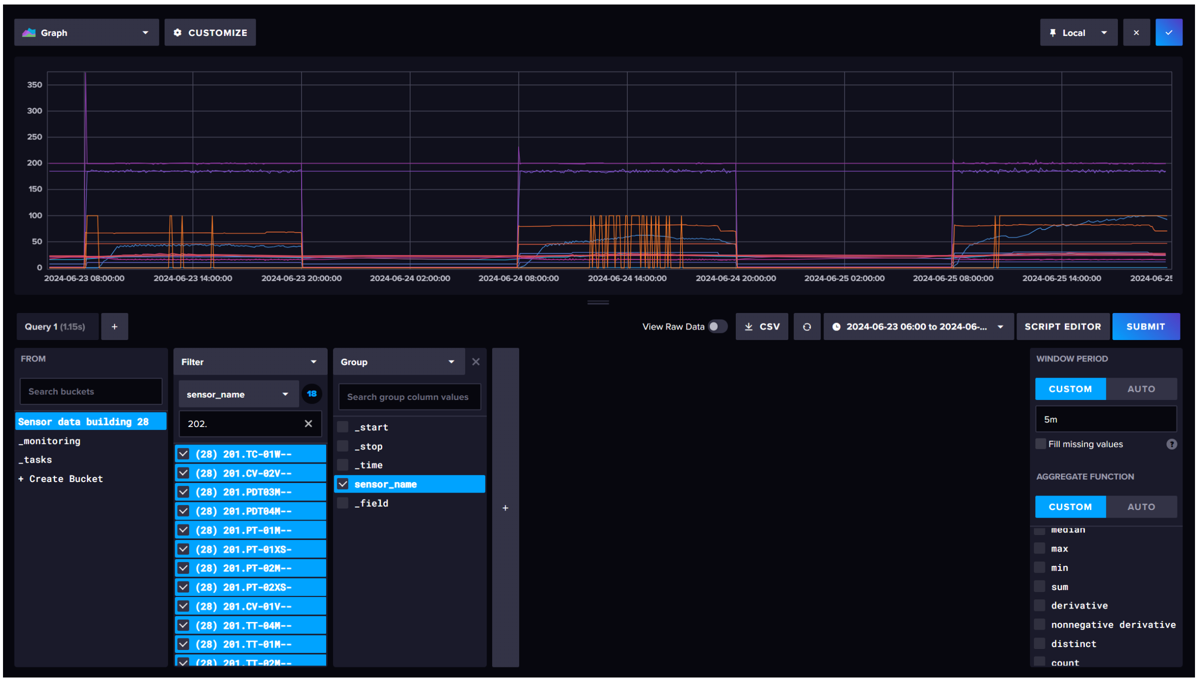


Figure 7.2: InfluxDB UI showing the case study sensor data collected from the Kafka cluster, from June 23rd to June 25th 2024.

The points stored in InfluxDB were queried for fault diagnosis with the query presented in Listing 7.2. The first row of the query determines the bucket from which to query data, this name is set as an environment variable. The second row controls the range of data gathered, here set between the 20th of June and the 27th of June. The 'aggregateWindow' function in the third row converts the points to 10-minute samples, by choosing the first sample in every 10-minute interval. The final row indexes the values by timestamp, with each sensor name as a column. Inputting the query to the 'query_api().query_data_frame' method results in a pandas DataFrame object formatted in this way. This DataFrame was preprocessed as described in Section 4.3.

```
for message in message_list:
    for measurement in message['measurements']:
        point = {
            "measurement": "sensor_data",
            "tags": {"building": message['building'], "sensor_name": measurement['
                measurement_id_name']},
            "time": message["timestamp"],
            "fields": {"value": measurements['value']}
        }
```

Listing 7.1: Python code used for converting Kafka messages with sensor data to InfluxDB points.

```
    QUERY_STRING = \
    f'from(bucket:"{os.getenv("INFLUX_BUCKET")}")␣' \
    f'|>␣range(start:␣2024-06-20T00:00:00Z,␣stop:␣2024-06-27T23:59:00Z)␣' \
    f'|>␣aggregateWindow(every:␣10m,␣fn:␣first,␣createEmpty:␣false)' \
```

```
    f'|>␣pivot(rowKey:["_time"],␣columnKey:␣["sensor_name"],␣valueColumn:␣"_value
        ")'
```

Listing 7.2: Flux query used for reading InfluxDB points.

7.3. Real-time diagnosis results

In this section, the result of applying the fault diagnosis method to the data gathered from the Kafka environment are shown.

Data was collected from Kafka between the 21st and 25th of June 2024. Some data was missing during this period, so the fault diagnosis is applied to a single day in this period to further validate and demonstrate the diagnosis method. The day chosen was the 24th of June 2024.

The setup for applying fault diagnostics was identical to the setup used for historical data. The faults were diagnosed in all 10-minute samples between 08:00 and 20:00, resulting in 72 samples. Compared to the historical data, additional sensors were available in 5-minute intervals for this dataset. However, for the sake of comparison to previous results, the diagnosis interval was maintained at 10 minutes. The daily symptom detection threshold was set to three samples and the consecutive symptom detection threshold to two samples. Therefore each symptom was again only considered if it was detected in two samples consecutively, more than three times per day. The revised DBN shown in B.1 was used for performing the diagnosis.

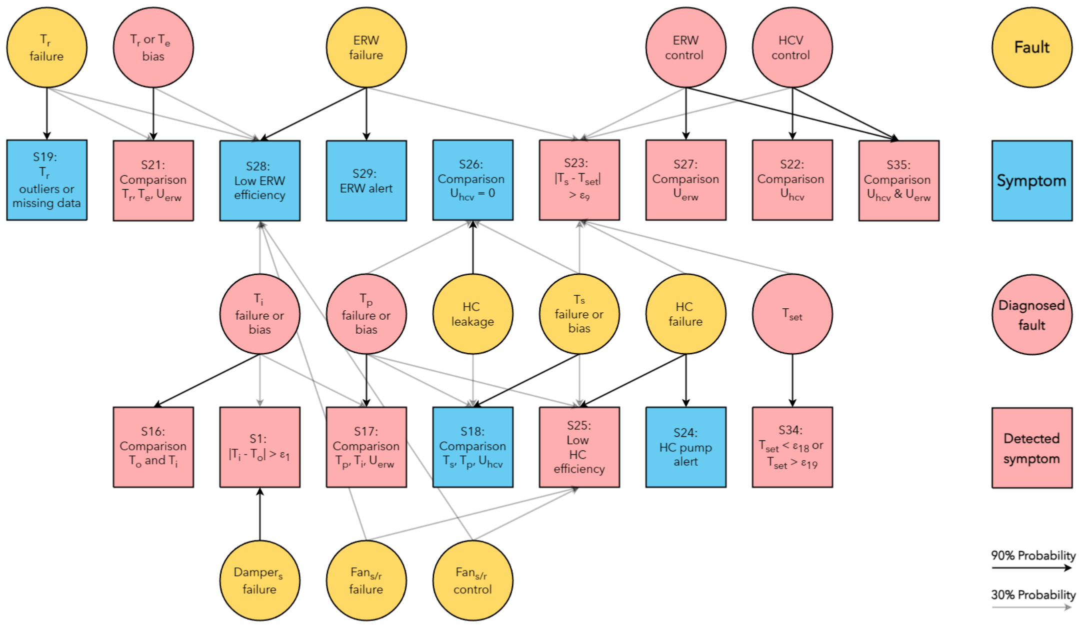


Figure 7.3: Symptoms detected and faults diagnosed for the 24th of June, shown in a part of the DBN.

In total, ten symptoms were detected and six faults were diagnosed, shown in the DBN in Figure 7.3. The percentages of detected samples and days are shown in Figure 7.4 for the detected symptoms, and in Figure 7.5 for the diagnosed faults. As the diagnosis was performed for a single day, the percentage of days considered or diagnosed was either 0% or 100%.

All of the symptoms and faults identified were related to temperature, and the majority were also frequently diagnosed in the historical dataset. However, this was not the case for symptoms S18 and S19, which were detected in the historical data but not in the real-time diagnosis. In contrast, symptom S21 was detected in the real-time diagnosis but not in the historical diagnosis.

With regards to the faults diagnosed, two faults were diagnosed differently compared to the historical data.

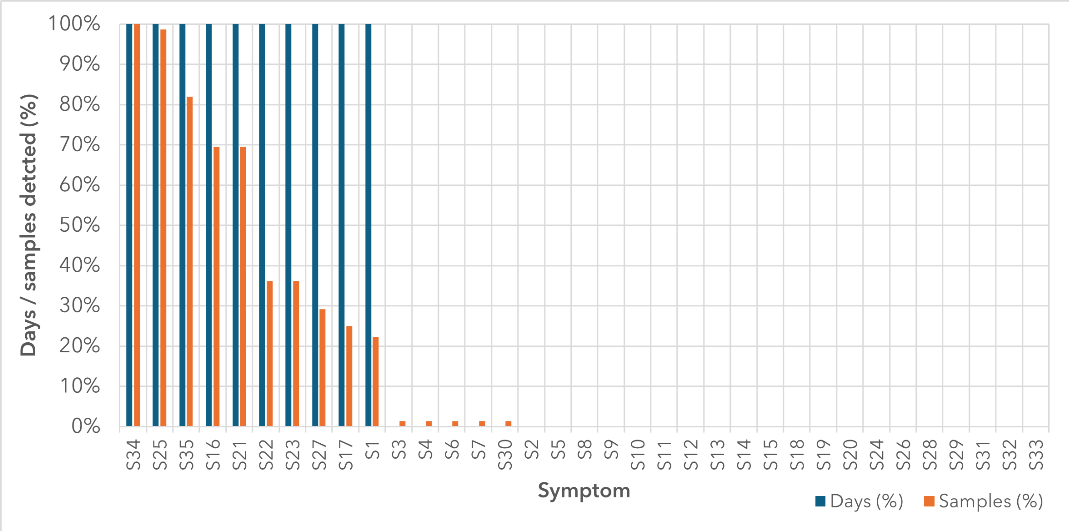


Figure 7.4: Symptoms detected with the revised DBN for the 24th of June 2024, based on data retrieved from the Kafka cluster.

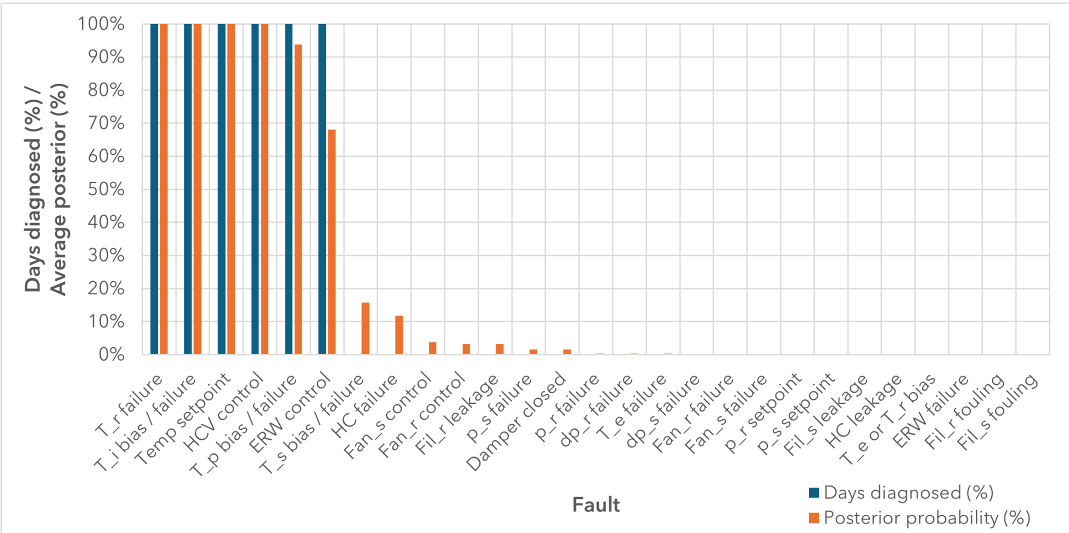


Figure 7.5: Faults diagnosed for the 24th of June 2024 with the revised DBN, based on data retrieved from the Kafka cluster.

These were the return temperature sensor failure and return or exhaust temperature sensor bias faults. The return temperature sensor failure fault was not diagnosed for the real-time data, as the sensor data was present. This fault was correctly diagnosed for all days in the historical dataset.

The return or exhaust temperature sensor bias fault was diagnosed for the real-time data, while it was not diagnosed for the historical data. This fault could not be diagnosed for the historical data due to the missing return temperature sensor data, in combination with the consecutive symptom threshold. Since the return temperature data was available in the real-time dataset, this fault could now be diagnosed.

The faults diagnosed were expected to align with the historical results, given that the operation of the case study AHU was not expected to have undergone any modifications. Furthermore, no alerts were present on the 24th of June, so no related component faults were expected to be present in the AHU. However, the 24th of June took place during the summer season, when the HC should not be operated. Therefore, the diagnosis of the HCV control fault is an unexpected outcome. The following figures show relevant sensor data from the 24th of June, explaining why this fault was diagnosed.

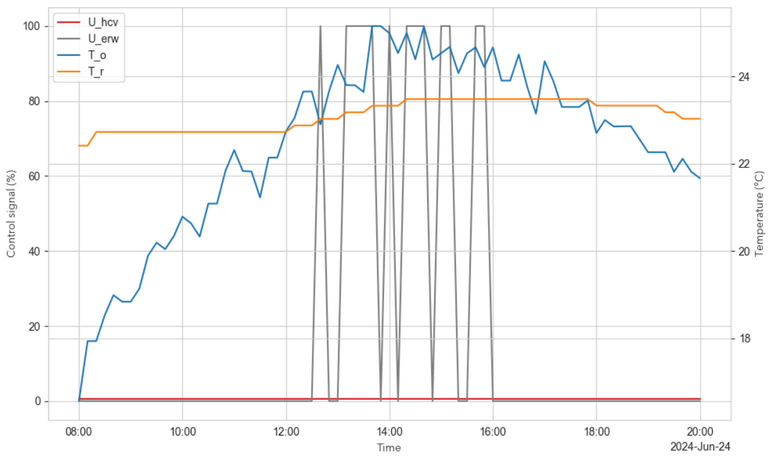


Figure 7.6: HCV and ERW control signals compared to temperature data for the 24th of June 2024.

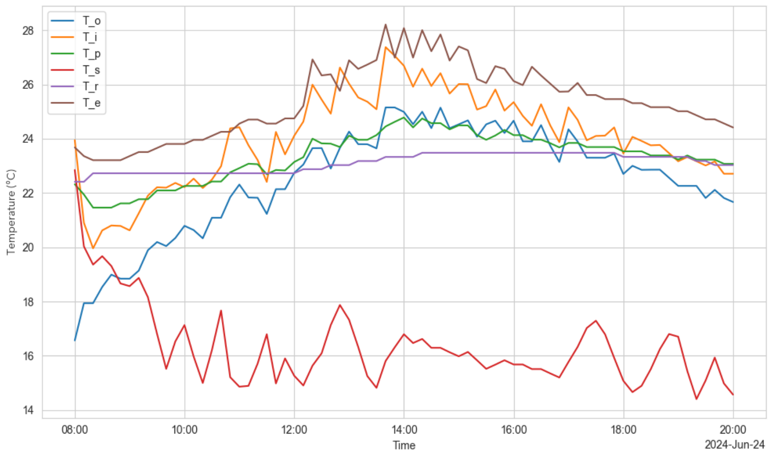


Figure 7.7: Temperature data for the 24th of June 2024.

Figure 7.6 shows the control signals of the HCV and ERW, as well as the outdoor and return temperatures. From this, we can determine whether symptoms S22, S25, S27 and S35, related to the HCV and ERW control, were detected correctly.

The HCV control signal shown in Figure 7.6 was set to 0.59%, while it should be 0% when the ERW is turned on. Therefore, sequencing symptom S35 was detected correctly. Furthermore, the supply temperature exceeded the setpoint by more than 1 °C for several samples, resulting in the accurate detection of HCV control symptom S22. This can be seen in Figure 7.7.

The symptom rule detecting low efficiency of the HC (S25) calculates the difference between the preheating and supply temperature when the HCV control signal is greater than 0%. Because the control signal is 0.59%, the efficiency is calculated even though the coil should not be used. The calculated efficiency was lower than the threshold, due to the cooling coil being used to provide cooling.

The ERW control symptom S27 is detected due to the ERW being turned on even though the return temperature is lower than the outdoor temperature. The ERW is used to recover cold, which is not in line with the control document of the AHU. If this is intentional, the symptom rules need to be adjusted to accommodate this. This is further discussed in Chapter 8.

Based on the outdoor and measured temperatures for the AHU, shown in Figure 7.7, the accuracy of the detected bias symptoms can be verified.

The outdoor and inlet temperatures follow a similar pattern, although the inlet temperature is about two degrees higher during most of the hours considered. Comparing the inlet and preheating temperature sensors, we would expect to see similar temperatures when the ERW is turned off, which is before 12:00 and after 16:00. However, when the inlet temperature increases around 11:00, following the increase in outdoor temperature, the preheating temperature does not increase as well. Similarly, the return and exhaust temperatures are expected to be similar when the ERW control signal is 0%. This is also not the case, as the exhaust temperature increases between 08:00 and 12:00, while the return temperature remains stable. Based on these temperatures, bias symptoms S1, S16 and S16 are accurately detected.

Finally, symptom S34 is also detected correctly due to the temperature setpoint being set beneath 19 °C. This can also be seen by the low supply temperature in Figure 7.7.

In total, six faults were diagnosed: bias of the return or exhaust temperature sensor; failure or bias of the inlet or preheating temperature sensor; an incorrect temperature setpoint; incorrect ERW control; and HC failure. Similar to the diagnosis of the historical data, the diagnosed faults correspond to the detected symptoms, as the faults with strong links were diagnosed. The only exceptions are the damper fault and HC failure faults, for which the inlet temperature sensor and preheating sensor faults were diagnosed instead. The diagnosis of the bias faults is further discussed in Chapter 8.

When performing a fault diagnosis for each sample in real-time, the amount of processing time needed for performing the fault diagnosis becomes relevant. For the conducted fault diagnosis on the 24th of June, the average processing time of performing fault diagnosis over ten runs was 9.2 seconds on an octa-core AMD Ryzen 7 7735HS with 16GB RAM. Including the time needed for loading the data from InfluxDB, the average over ten runs was 9.8 seconds. Therefore, running the fault diagnosis for every 5- or 10-minute sample seems to be feasible. In practice, more time would be required to retrieve the data from Kafka and write to InfluxDB. The processing time needed for retrieving the data from Kafka and writing this to InfluxDB depends on the number of messages that are sent during the time interval.

8
Discussion

This chapter discusses the results from the previous chapter. The results of the experimental, historical and real-time data are discussed. In addition, the generalisability of the proposed method is discussed. The chapter concludes with an analysis of the implications on sustainability of the research.

8.1. Experimental results

The experiments showed that the DBN was able to correctly diagnose most introduced faults with all considered diagnosis periods, except for the ERW control fault, the supply temperature sensor fault and the pressure sensor fault. For the ERW control fault, a separate symptom was introduced in the revised DBN presented in Appendix B. With this additional symptom, the DBN was able to correctly diagnose the ERW control fault in all diagnosis periods.

The supply temperature sensor fault could not be diagnosed because the HCV was opened during the experiment. If the DBN is extended to also diagnose outside of operation hours, when the HCV is closed, the fault can be diagnosed. During operation, the fault could also be diagnosed if the supply water temperature sensor at the heating coil was present, so the heat transfer could be calculated and compared to the measured supply temperature.

The pressure sensor fault was not included in the DBN, due to missing sensors in the case study AHU. However, a pressure sensor bias experiment was still conducted, to find how reliable the diagnosis method is when faults occur that are not included. Although the DBN still diagnosed faults, a fan control fault was diagnosed when the pressure sensor value was set to 200 Pa. Since this fault could not have been present during the experiment, considering the control signal of the fan was 0%, the DBN structure should be altered to include the pressure sensor fault as well.

For the four shortest diagnosis periods, the HCV control fault was diagnosed instead of the supply temperature fault. Only in the longest diagnosis period of three hours, the HC failure fault was diagnosed, which is more similar to the supply temperature sensor fault than the HCV control fault. The 2-hour and 3-hour diagnosis periods both diagnosed the same samples, the only difference was the total number of symptoms threshold. This shows that the thresholds enabled the DBN to improve the diagnosis of this fault.

Pressure sensor faults were not included in the DBN, so the pressure sensor faults during the experiments were diagnosed as damper or fan control faults instead. A fan rotation speed or power consumption sensor, included in the DBN by Zhao et al., could be used to diagnose this fault [62]. Symptoms could also be added to detect frozen sensor values by comparing the sensor values of multiple samples. This was not done for the proposed diagnosis method, so the diagnosis could be applied for a one-sample period as well. Although the faults were not included in the proposed DBN, the developed method was able to detect faulty operation even though the pressure faults were not included. Therefore, other faults that were also not included could still lead to a fault diagnosis.

8.2. Historical results

The diagnosis of 2022 and 2023 found control and sensor faults during normal operation of the AHU for a large portion of the dataset. No faults were diagnosed for the other components, which is in line with the lack of failure alerts in the diagnosed period. The most notable detected symptom was symptom S23, which was detected in 79% of the days. This symptom is detected if the difference between the supply temperature and temperature setpoint is greater than ε_9, set to 1 °C. The most likely cause for this symptom being detected is incorrect control of the HCV and ERW, as symptoms S22 and S27 were also detected often, indicating incorrect control signals.

All of the supply-side temperature sensors were frequently diagnosed to be biased. As it is unlikely that all of the temperature sensors were biased, the symptoms related to the sensor faults could be detected too often because of too sensitive thresholds. For example, the symptom strongly linked to the inlet sensor (S16) was detected if the difference between the inlet temperature sensor and outdoor temperature data was greater than ε_7, which was set to 1 °C. Perhaps this temperature difference is often larger than 1 °C in practice, due to a difference between the outdoor temperature data and the actual temperature at the location of the AHU. For the biases between inlet, preheating and supply temperatures, and return and exhaust temperatures the threshold values could also be increased to avoid false positive results. A different approach for detecting these biases would be to compare the temperatures only outside of operation hours when no heat transfer takes place within the AHU.

All of the symptoms that are detected in more than 5% of the samples are temperature-related. The pressure setpoints are constant in virtually all of the samples, although the return pressure setpoint has been increased from 185 Pa to 200 Pa. It is assumed that this increase is intentional, but this should be investigated with building management. Airflow sensors or rotation speed sensors could be used to detect additional pressure-related symptoms, which could find lower fan efficiency faults. Sensor bias could then also be diagnosed separately. The low number of samples in which symptoms S4 and S7 are detected shows that the AHU provides the requested setpoint pressure for most of the samples.

The missing data of the return temperature sensor is correctly diagnosed for all of the samples with a calculated posterior fault probability of 100%. This shows the ability of the developed method to diagnose missing sensor data faults, which are expected to apply to the other sensors as well.

The consecutive symptom threshold in combination with missing data caused some symptoms to be unable to be diagnosed. Therefore, when interpreting the results of the developed method it should be considered that if a sensor failure is diagnosed, some other faults cannot be diagnosed due to missing data. In the historical data, this was the case for the return temperature sensor, making it impossible to detect low ERW efficiency in multiple samples consecutively. This was also the case for other rules that included the return temperature, such as symptom S21, which detects a bias between the return and exhaust temperatures.

If other sensors are missing, the diagnosed faults would be affected in the same manner. This would be problematic for the supply temperature sensor in particular, as this value is used in multiple other symptoms. Therefore, if the diagnosis results often include a sensor, the consequences for other faults should be taken into consideration. For example, in the case of the supply temperature sensor, the symptom signalling a deviation from the setpoint would then not be detected.

8.3. Real-time results

The real-time diagnosis showed several findings that can be used to further adapt the DBN for the case study AHU.

Firstly, the results of the real-time diagnosis showed that the HCV valve was not fully closed during the summer season. However, the value was 0.59% instead of 0%, so this fault could be considered insignificant. Additionally, the boilers should not provide warm water to the HC during the summer. The symptom rules related to the HC can be altered depending on the desired sensitivity by the building operators. For example, a threshold could be introduced in the threshold rules, if the control signal is higher than 5% instead of 0%. Furthermore, if the HCV control signal is irrelevant during the summer season, the current season can be used as a variable for these symptom detection rules. Then, incorrect HCV control and ERW sequencing would only be diagnosed during the winter and in-between seasons.

Another aspect that should be noted related to the HCV signal, is that this may have impacted the historical analysis as well. If the HCV control signal was set slightly above 0% during a large part of the dataset, this may have been part of the reason why this fault was frequently diagnosed.

Another result of the real-time diagnosis was that the ERW is also used for cold recovery, which was not listed in the design document of the AHU. If this is intentional, the symptom detection rule of ERW control symptom S27 should be adapted.

Finally, comparing the temperature sensors showed that the differences between the temperature sensors were greater than previously expected. Particularly for the return and exhaust temperature sensors, when the ERW was not used, the exhaust temperature still deviated from the return temperature. The return temperature remained stable throughout the day, only increasing about 1 °C. The exhaust temperature followed the trend of the outdoor and inlet temperature sensors, increasing about 4 °C.

Although multiple sensors in the case study AHU could have been biased, this is quite unlikely. A more reasonable explanation is that the temperatures during operation can deviate more than 1 °C, and that therefore the threshold should be increased. Another approach could be to instead compare the temperatures only outside of operation hours.

8.4. Generalisability of the diagnosis method

As mentioned previously, several faults could be added to the DBN if additional sensors are available. Bias or frozen values of the pressure sensors can be diagnosed by measuring the power consumption or rotation speed of the fans. A temperature sensor at the heating coil water supply can be used to calculate the expected heat transfer for the coil, improving the accuracy of diagnosing heating coil failure. Finally, a supply airflow sensor could be used to diagnose low fan efficiency. However, these sensors are often not available in practice and therefore DBNs need to be developed that are generalisable to most AHUs.

The DBN formulated in this thesis is expected to contain sensors available in most AHUs. However, the diagnosis accuracy is improved by integrating BMS alerts as symptoms. This could potentially reduce the generalisability of the method if other AHUs do not have this information available within the BMS, or if the components have no failure alerts. Because all of the symptom rules are knowledge-based, the method does not require historical data to perform the diagnosis. The thresholds used can be changed depending on expert knowledge of the AHU, however. The diagnosis period can also be altered as needed, ranging from single samples to an entire day.

Additionally, the case study AHU maintains a constant pressure setpoint. The symptom rules for the pressure setpoints would need to be adapted for AHUs that control the setpoint based on occupancy. However, the structure of the DBN would not need to be altered.

The approach to adapting the DBN for other AHUs with an ERW is relatively straightforward. Faults for any components that are absent can be removed, as well as any edges from these faults. Symptoms with detection rules that include any absent sensors can also be removed. In some cases, the calculation can be adapted, such as for the ERW efficiency. This symptom rule currently calculates the efficiency based on the return and exhaust temperatures, which could also be calculated with the inlet and preheating temperatures. Depending on which faults and symptoms remain, the conditional probabilities between symptoms and faults can be set appropriately.

The DBN structure would have to be altered more significantly to perform accurate diagnosis of AHUs without an ERW. This is caused by the mixing box, which both the supply and return airflow pass through. Therefore, the biases can not be calculated by comparing the return and exhaust temperature sensors, as well as the inlet and preheating temperatures. The ERW control fault and symptom could be replaced with the mixing damper control fault and a symptom comparing the control signal to intended values. A stuck mixing damper could be diagnosed with a symptom that compares the temperature sensors around the mixing damper, instead of the low ERW efficiency symptom.

8.5. Sustainability implications

To estimate the increased energy usage of the faults introduced during the experiments, the energy usage of the building during the experiments was compared to a day with a similar outdoor temperature. A month before the experiments were conducted, the average outdoor temperature was similar. Instead of 6.9 °C the average outdoor temperature was 6.1 °C. Comparing the energy usage between these two days showed that the gas usage was about half during the experiments, even though faults were introduced in the AHU. Therefore this approach does not give a correct estimation of energy waste caused by faults in the AHU. This was most likely due to lower occupancy during the experiments, as they were conducted during weekends when no occupants were present.

In the analysis of the historical data, most of the diagnosed faults are control and sensor faults, which can remain unnoticed without performing FDD. The return, inlet and preheating sensors were often diagnosed as faulty. With regards to the energy performance of the AHU, the preheating and inlet temperature sensors do not have a strong impact on energy usage as the values are not used to control any components. They are needed for performing fault diagnosis of the AHU, however. The return temperature sensor is used for controlling the ERW, so this can have an impact on energy usage. Since the sensor was only available every 30 minutes, the control could have been sub-optimal. However, it can also be the case that the data was available in real-time, but not stored in 10-minute intervals.

The temperature setpoint fault was also diagnosed for a large portion of the dataset. This was mainly because the setpoint was lower than the intended values of the AHU. Setting the setpoint lower decreases the energy usage of the AHU, however, it may also have an impact on the other HVAC systems in the building. For example, by delivering colder air to the building the radiators may be needed to supply more heat instead, which could increase gas consumption. This is difficult to verify, as the energy consumption of the AHU has not been measured.

Additional faults that have an impact on the energy usage of the AHU are the HCV and ERW controls. Energy can be wasted when the HCV control signal is too high for the current supply temperature. The AHU also opens the HCV while the ERW is turned off, which does not maximise the available energy recovery potential. Additionally, the ERW control signals can only turn the ERW on and off, which is less efficient than staying at a constant capacity.

If the ERW is not operated in the summer season, as stated in the design document, this also reduces the amount of energy that can be recovered. However, Figure 7.6 in the previous chapter shows that the ERW is also used to recover cold. Therefore it is likely that the design documents are incorrect.

9 Conclusion

This chapter concludes this thesis by answering the research question. It also offers a reflection on the chosen methods and limitations of the conducted research. Finally, possibilities for future research are described.

This thesis aims to provide a general fault diagnosis method that can be used for real-time diagnosis of AHU faults. The main research question of this thesis was therefore formulated as the following:

'How can Bayesian network-based fault diagnosis accurately find faults for air-handling units in real-time based on expert knowledge?'

In order to answer the main research question, the sub-questions can now be answered:

'How can expert knowledge be transferred to a diagnostic Bayesian network to detect and diagnose common air-handling unit faults?'

Common AHU faults from the literature were presented in Chapter 4.1. Symptoms rules were formulated based on these faults, by applying the 4S3F framework. This was done with a knowledge-based approach, based on design documents of the AHU and conversations with building management.

The final DBN is presented in Appendix B. The DBN was shown to be able to diagnose AHU control faults successfully. Additional validation is needed for component faults, however. The symptom rules presented can also be applied to other AHUs with an ERW and similar sensors. Furthermore, a similar approach can be used to formulate the symptom detection rules and DBN structure.

'Which faults are diagnosed with Bayesian network-based fault detection and diagnosis during normal operation of air-handling units?'

In Chapter 6, the historical data was diagnosed with the developed diagnosis method. The faults found most frequently were all temperature-related. Three significant faults were diagnosed for a large number of days: an incorrect temperature setpoint (64% of the dataset), incorrect HCV control (43%) and incorrect ERW control (11%). These faults are expected to have impacted the energy consumption of the AHU.

Besides sensor-related faults, no component faults were diagnosed with the developed method for the case study. This aligned with the lack of alerts in the BMS for the components in the diagnosed period.

'How can Bayesian network-based fault detection and diagnosis be applied to diagnose air-handling unit faults in real-time?'

In Chapter 5, experiments were introduced where faults were diagnosed in various diagnosis intervals. The results have shown that the revised DBN was able to diagnose all control faults even using only a 1-sample detection period. However, using more samples is recommended, as the symptom thresholds helped improve the accuracy of the diagnosis for longer periods.

In Chapter 7 diagnosis of data gathered through the data streaming platform Apache Kafka was performed. Data was collected from Kafka and saved locally in InfluxDB, a time-series storage solution. The required timespan was queried so diagnosis could be performed.

This showed that real-time diagnosis could be performed with a data streaming framework. The measured processing time of performing diagnosis enables fault diagnosis to be performed on every 5- or 10-minute sample.

By answering the sub-research questions, the main research question is also answered. The developed method has been shown to be able to detect and diagnose real-time data, by transferring expert knowledge to a DBN. The following sections describe the main limitations and future work that can be undertaken to further improve and validate the method.

9.1. Limitations

In this section, the main limitations of the research are described which should be considered for the presented results. For the historical fault diagnosis, some data was not fully available. The return temperature sensor was only available in 30-minute intervals from September 2022. Due to the consecutive symptom thresholds, the symptoms using this sensor could not be detected. Additionally, historical alerts were not available for 2023, so it was assumed that no alerts were present in that year. Because of these limitations, some component faults could not have been diagnosed.

The second aspect that should be considered is the effect of missing airflow and HC water supply temperature sensors. Because these sensors were absent, the efficiencies of the HC and fans could not be estimated. Therefore faults related to fan efficiency were excluded. Additionally, the missing sensors could impact the reliability of the fan failure and HC failure fault diagnoses.

The third limitation is related to the chosen threshold values used in the symptom detection rules. Several proposed epsilon values were based on estimations. Some of these values may have been too sensitive, especially for biases of the temperature sensors. Additionally, the values were based on control and technical documents of the AHU. This information could be incorrect or outdated, for example, the fan efficiency can be lower in practice than stated in the design document. Furthermore, technical information could be lacking when applying the diagnosis method to other AHUs. This highlights the importance of involving building operators in the setup of the diagnosis method, as they can provide input on sensible threshold values for the considered AHU.

Another limitation of the results is that they are dependent on the conditional and prior probabilities of the DBN. The sensitivity analysis of Section 5.5 showed that the structure affected the number of false positive and negative faults diagnosed.

Finally, some symptoms could be detected due to the transient nature of the data. There is a delay between changes in the control signals and the measured temperatures. For example, when the ERW is turned on just before the sampling interval, the supply temperature may still lie below the setpoint. This should be considered when setting the thresholds in the symptom rules. Alternatively, future work can be done on filtering transient data, which is discussed further in the next section.

9.2. Recommendations for future work

This section will outline recommended future work related to this thesis.

The first recommended future research is to extend the symptoms and faults of the DBN. The DBN can be extended to the summer season by adding symptoms and faults related to the chiller. ERW control during the summer could also be further validated. Secondly, diagnostics can be performed outside of working hours, when only temperature sensor distortions are detected. This is expected to improve the accuracy of the bias-related symptoms.

The symptoms could also be improved by adding efficiency estimations for the coils and fans based on engineering knowledge. The expected efficiency could then be calculated based on the control signals. Symptoms that cover several samples could also be added, such as the freezing of sensor values symptom. For this thesis, only symptoms that can be detected in real-time with one sample have been considered.

The proposed epsilon values should also be further investigated. This could be done by conducting data-driven research for different AHUs and by interviewing engineers with experience in operating AHUs. Applying the diagnostic method to different case study AHUs can also further validate the detection rules.

The final suggestion related to symptom detection is to test different methods of filtering transient data. Zhao et al. filter the data based on the minimum, maximum and mean sensor values of the previous five samples, as proposed by Lee et al [31, 62]. If this is done for real-time diagnosis, a trade-off must be found so that the amount of data left is sufficient to perform real-time fault diagnosis. Additionally, the combination of filtering the transient data and only selecting symptoms with the total and consecutive thresholds can be explored.

The second area for future research is the structure of the DBN. The current implementation of the DBN contains binary states for the symptom and fault nodes, absent and present. The number of states can be extended, e.g. to positive, negative and fault-free states. This allows separate conditional probabilities for the different states, potentially improving the diagnosis accuracy.

Similarly, the number of parameters for the conditional probabilities could be extended. For example, a medium strength conditional probability parameter could be added of 60%. Furthermore, different parameter values can be experimented with.

Different values for the posterior threshold of 60% can be further investigated for the fault diagnosis process. As an alternative approach, faults could be isolated similar to the method proposed by Zhao et al [62]. In their work, faults are considered if they are at least some threshold higher than the other faults (for example 30%). Additionally, if a fault is not isolated a fault alert is still generated if the air pressure and fan power consumption exceed certain thresholds.

The real-time diagnosis presented in Chapter 7 can be further extended by adding alerts from the BMS to the data streaming platform. For the case study, the alerts are stored differently than sensor data, which could make integration with the real-time framework challenging. Furthermore, the alerts are only stored for one sample in the BMS, even though the related fault could be present for a longer period. Therefore, each alert should be considered for multiple samples.

The developed Python implementation can be further improved as well. Firstly, the code can be run on a server to test automated diagnosis. Furthermore, a graphical user interface can be implemented with a framework such as Dash, to provide diagnosis information on an accessible website [42]. This information could consist of showing the DBN, as well as the diagnosis results with detected symptoms, diagnosed faults and calculated posteriors for the period considered. Additionally, the developed method should be applied to different AHUs in buildings on the TU Delft campus and elsewhere, to research the generalisability of the method. The most logical first candidate is the case study building, which contains another AHU.

Finally, additional experiments can be performed to simulate component failures and further validate the proposed DBN. During the experiments, alarms should be generated which can be analysed to determine the frequency of false positives or false negatives. In addition, the energy consumption during these experiments can be measured and compared to fault-free operation.

References

[1] Muhammad Waseem Ahmad, Monjur Mourshed, Baris Yuce, and Yacine Rezgui. "Computational intelligence techniques for HVAC systems: A review". In: *Building Simulation 2016 9:4* 9.4 (Mar. 2016), pp. 359–398. ISSN: 1996-8744. DOI: 10.1007/S12273-016-0285-4. URL: https://link.springer.com/article/10.1007/s12273-016-0285-4.

[2] Kamilla Heimar Andersen, Simon Pommerencke Melgaard, Hicham Johra, Anna Marszal-Pomianowska, Rasmus Lund Jensen, and Kvols Heiselberg. "Barriers and drivers for implementation of automatic fault detection and diagnosis in buildings and HVAC systems: An outlook from industry experts". In: *Energy & Buildings* 303 (2024), p. 113801. DOI: 10.1016/j.enbuild.2023.113801.

[3] Bharathan Balaji, Arka Bhattacharya, Gabriel Fierro, Jingkun Gao, Joshua Gluck, Dezhi Hong, Aslak Johansen, Jason Koh, Joern Ploennigs, Yuvraj Agarwal, Mario Berges, David Culler, Rajesh Gupta, Mikkel Baun Kjærgaard, Mani Srivastava, and Kamin Whitehouse. "Brick: Towards a unified metadata schema for buildings". In: *Proceedings of the 3rd ACM Conference on Systems for Energy-Efficient Built Environments, BuildSys 2016*. Association for Computing Machinery, Inc, Nov. 2016, pp. 41–50. ISBN: 9781450342643. DOI: 10.1145/2993422.2993577.

[4] Brains4Buildings. *Brains4Buildings.org*. 2021. URL: https://brains4buildings.org/.

[5] Baoping Cai, Yonghong Liu, Qian Fan, Yunwei Zhang, Zengkai Liu, Shilin Yu, and Renjie Ji. "Multi-source information fusion based fault diagnosis of ground-source heat pump using Bayesian network". In: *Applied energy* 114 (2014), pp. 1–9.

[6] Samuel Caillou and Paul Van den Bossche. "Heat recovery efficiency: measurement and calculation methods". In: 2012. URL: www.bbri.be.

[7] Jianli Chen, Liang Zhang, Yanfei Li, Yifu Shi, Xinghua Gao, and Yuqing Hu. "A review of computing-based automated fault detection and diagnosis of heating, ventilation and air conditioning systems". In: *Renewable and Sustainable Energy Reviews* 161.April (2022), p. 112395. ISSN: 18790690. DOI: 10.1016/j.rser.2022.112395. URL: https://doi.org/10.1016/j.rser.2022.112395.

[8] Chitkara. *Continuous Monitoring and Automated Fault Detection and Diagnosis of Large Air-Handling Units*. Tech. rep. 2022. URL: www.tue.nl/taverne.

[9] Shobhit Chitkara, Alet Van Den Brink, Shalika Walker, and Wim Zeiler. "An early prototype for fault detection and diagnosis of Air-Handling Units". In: *Proceedings of the REHVA 14 HVAC World Congress* (2022), pp. 1–8. DOI: 10.34641/clima.2022.318. URL: https://doi.org/10.34641/clima.2022.318.

[10] Confluent. *Confluent Kafka Python*. 2024. URL: https://github.com/confluentinc/confluent-kafka-python.

[11] Suhrid Deshmukh, Stephen Samouhos, Leon Glicksman, and Leslie Norford. "Fault detection in commercial building VAV AHU: A case study of an academic building". In: *Energy and Buildings* 201 (Oct. 2019), pp. 163–173. ISSN: 0378-7788. DOI: 10.1016/J.ENBUILD.2019.06.051.

[12] Francisco J Díez and Marek J Druzdzel. *Canonical Probabilistic Models for Knowledge Engineering*. Tech. rep.

[13] European Commission. "Commission Regulation (EU) No 1253/2014 of 7 July 2014 implementing Directive 2009/125/EC of the European Parliament and of the Council with regard to ecodesign requirements for ventilation units". In: *Official Journal of the European Union* (2014).

[14] European Parliament. “Directive (EU) 2024/1275 of the European Parliament and of the Council of 24 April 2024 on the energy performance of buildings ”. In: *Official journal of the European Union* (2024). URL: `http://data.europa.eu/eli/dir/2024/1275/oj`.

[15] Eurovent. *Air Leakages in Air Handling Units: Guidelines for Improving Indoor Air Quality and Correcting Performance*. Tech. rep. Brussels, Feb. 2021.

[16] *GitHub - influxdata/influxdb-client-python: InfluxDB 2.0 python client*. URL: `https://github.com/influxdata/influxdb-client-python`.

[17] Srinivasan Gopalan, Agata Rijs, Shobhit Chitkara, Anand Thamban, and Rick Kramer. “Fault prioritisation for Air Handling units using fault modelling and actual fault occurrence data”. In: *Energy and Buildings* 319 (Sept. 2024), p. 114476. ISSN: 0378-7788. DOI: `10.1016/J.ENBUILD.2024.114476`.

[18] Jessica Granderson, Guanjing Lin, Rupman Singla, Ebony Mayhorn, Paul Ehrlich, Draguna Vrabie, and Stephen Frank. “Commercial fault detection and diagnostics tools: what they offer, how they differ, and what’s still needed”. In: (2018).

[19] B. Gunay, B.W. Hobson, D. Darwazeh, and J. Bursill. “Estimating energy savings from HVAC controls fault correction through inverse greybox model-based virtual metering”. In: *Energy and Buildings* 282 (2023). DOI: `10.1016/j.enbuild.2023.112806`.

[20] Burak Gunay, Weiming Shen, and Chunsheng Yang. “Characterization of a building’s operation using automation data: A review and case study”. In: *Building and Environment* 118 (June 2017), pp. 196–210. ISSN: 0360-1323. DOI: `10.1016/J.BUILDENV.2017.03.035`.

[21] Anna Heidtmann. *Exploring the integration of occupant behaviour in fault detection and diagnosis in smart buildings*. Tech. rep.

[22] Min Hu, Huanxin Chen, Limei Shen, Guannan Li, Yabin Guo, Haorong Li, Jiong Li, and Wenju Hu. “A machine learning bayesian network for refrigerant charge faults of variable refrigerant flow air conditioning system”. In: *Energy and Buildings* 158 (2018), pp. 668–676. ISSN: 03787788. DOI: `10.1016/j.enbuild.2017.10.012`. URL: `http://dx.doi.org/10.1016/j.enbuild.2017.10.012`.

[23] *InfluxDB data elements | InfluxDB OSS v2 Documentation*. URL: `https://docs.influxdata.com/influxdb/v2/reference/key-concepts/data-elements/`.

[24] *Install InfluxDB | InfluxDB OSS v2 Documentation*. URL: `https://docs.influxdata.com/influxdb/v2/install/?t=Windows+Powershell`.

[25] Asad Javed, Jérémy Robert, Keijo Heljanko, and Kary Främling. “IoTEF: A Federated Edge-Cloud Architecture for Fault-Tolerant IoT Applications”. In: *Journal of Grid Computing* 18.1 (Mar. 2020), pp. 57–80. ISSN: 15729184. DOI: `10.1007/S10723-019-09498-8`. URL: `https://www.researchgate.net/publication/338516815_IoTEF_A_Federated_Edge-Cloud_Architecture_for_Fault-Tolerant_IoT_Applications`.

[26] Finn V Jensen. *An introduction to Bayesian networks*. Vol. 210. UCL press London, 1996.

[27] Srinivas Katipamula and Michael R Brambley. “HVAC&R Research Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems-A Review, Part I”. In: 11.1 (2005), pp. 3–25. ISSN: 1938-5587. DOI: `10.1080/10789669.2005.10391123`. URL: `https://www.tandfonline.com/action/journalInformation?journalCode=uhvc21`.

[28] Srinivas Katipamula and Michael R Brambley. “HVAC&R Research Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems-A Review, Part II”. In: 11.2 (2005), pp. 169–187. ISSN: 1938-5587. DOI: `10.1080/10789669.2005.10391133`.

[29] Woohyun Kim and Srinivas Katipamula. "A review of fault detection and diagnostics methods for building systems". In: *Science and Technology for the Built Environment* 24.1 (Jan. 2018), pp. 3–21. ISSN: 2374474X. DOI: 10.1080/23744731.2017.1318008.

[30] w Lawrance and T Schreck. "Rotary heat exchangers save energy and prevent a need for recirculation which contributes to the decrease the risk of COVID-19 transfe". In: *REHVA Journal* (2020).

[31] Won Yong Lee, John M. House, and Nam Ho Kyong. "Subsystem level fault diagnosis of a building's air-handling unit using general regression neural networks". In: *Applied Energy* 77.2 (Feb. 2004), pp. 153–170. ISSN: 0306-2619. DOI: 10.1016/S0306-2619(03)00107-7.

[32] Tingting Li, Yang Zhao, Chaob Zhang, and Xuejun Zhang. "A knowledge-guided and data-driven method for building HVAC systems fault diagnosis". In: *Building and Environment* 198.March (2021). ISSN: 03601323. DOI: 10.1016/j.buildenv.2021.107850.

[33] Yijun Luo. "Fault Detection and Diagnosis of Temperature Sensors of Air Handling Unit Systems with Diagnostic Bayesian Networks". PhD thesis. 2022. URL: https://pure.tue.nl/ws/portalfiles/portal/198095619/1417487_master_thesis_report.Luo.pdf.

[34] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[35] Simon P. Melgaard, Kamilla H. Andersen, Anna Marszal-Pomianowska, Rasmus L. Jensen, and Per K. Heiselberg. "Fault Detection and Diagnosis Encyclopedia for Building Systems: A Systematic Review". In: *Energies 2022, Vol. 15, Page 4366* 15.12 (June 2022), p. 4366. ISSN: 1996-1073. DOI: 10.3390/EN15124366. URL: https://www.mdpi.com/1996-1073/15/12/4366/htm%20https://www.mdpi.com/1996-1073/15/12/4366.

[36] Daniel Minoli, Kazem Sohraby, and Benedict Occhiogrosso. "IoT Considerations, Requirements, and Architectures for Smart Buildings-Energy Optimization and Next-Generation Building Management Systems". In: *IEEE Internet of Things Journal* 4.1 (Feb. 2017), pp. 269–283. ISSN: 23274662. DOI: 10.1109/JIOT.2017.2647881.

[37] Maryam Sadat Mirnaghi and Fariborz Haghighat. "Fault detection and diagnosis of large-scale HVAC systems in buildings using data-driven methods: A comprehensive review". In: *Energy and Buildings* 229 (Dec. 2020), p. 110492. ISSN: 0378-7788. DOI: 10.1016/J.ENBUILD.2020.110492.

[38] Massieh Najafi, David M. Auslander, Peter L. Bartlett, Philip Haves, and Michael D. Sohn. "Application of machine learning in the fault diagnostics of air handling units". In: *Applied Energy* 96 (Aug. 2012), pp. 347–358. ISSN: 0306-2619. DOI: 10.1016/J.APENERGY.2012.02.049.

[39] Mohammad Nasar and Mohammad Abu Kausar. "Suitability Of Influxdb Database For IoT Applications". In: *Article in International Journal of Innovative Technology and Exploring Engineering* 10 (2019), pp. 2278–3075. DOI: 10.35940/ijitee.J9225.0881019. URL: https://www.researchgate.net/publication/335542829.

[40] Ondřej Nehasil, Lucie Dobiášová, Vojtěch Mazanec, and Jan Široký. "Versatile AHU fault detection – Design, field validation and practical application". In: *Energy and Buildings* 237 (Apr. 2021), p. 110781. ISSN: 0378-7788. DOI: 10.1016/J.ENBUILD.2021.110781.

[41] Luis Pérez-Lombard, José Ortiz, and Christine Pout. "A review on buildings energy consumption information". In: *Energy and Buildings* 40.3 (Jan. 2008), pp. 394–398. ISSN: 0378-7788. DOI: 10.1016/J.ENBUILD.2007.03.007.

[42] Plotly. *GitHub - plotly/dash: Data Apps & Dashboards for Python. No JavaScript Required.* URL: https://github.com/plotly/dash.

[43] Ojas Pradhan, Jin Wen, Yimin Chen, Xing Lu, Mengyuan Chu, Yangyang Fu, Zheng O'neill, Teresa Wu, and K Selcuk Candan. "Dynamic Bayesian Network-Based Fault Diagnosis for ASHRAE Guideline 36: High Performance Sequence of Operation for HVAC Systems". In: (2021). DOI: 10.1145/3486611.3491124. URL: https://doi.org/10.1145/3486611.3491124.

[44] Jeffrey Schein, Steven T. Bushby, Natascha S. Castro, and John M. House. "A rule-based fault detection method for air handling units". In: *Energy and Buildings* 38.12 (Dec. 2006), pp. 1485–1492. ISSN: 0378-7788. DOI: 10.1016/J.ENBUILD.2006.04.014.

[45] Arie Taal and Laure Itard. "Fault detection and diagnosis for indoor air quality in DCV systems: Application of 4S3F method and effects of DBN probabilities". In: *Building and Environment* 174.December 2019 (2020), p. 106632. ISSN: 03601323. DOI: 10.1016/j.buildenv.2019.106632.

[46] Arie Taal and Laure Itard. "P&ID-based automated fault identification for energy performance diagnosis in HVAC systems: 4S3F method, development of DBN models and application to an ATES system". In: *Energy and Buildings* 224 (2020), p. 110289. ISSN: 03787788. DOI: 10.1016/j.enbuild.2020.110289. URL: https://doi.org/10.1016/j.enbuild.2020.110289.

[47] Arie Taal and Laure Itard. "P&ID-based symptom detection for automated energy performance diagnosis in HVAC systems". In: *Automation in Construction* 119.May (2020), p. 103344. ISSN: 09265805. DOI: 10.1016/j.autcon.2020.103344. URL: https://doi.org/10.1016/j.autcon.2020.103344.

[48] Arie Taal, Laure Itard, and Wim Zeiler. "A diagnostic Bayesian network method to diagnose building energy performance". In: *Building Simulation Conference Proceedings* 2 (2019), pp. 893–899. ISSN: 25222708. DOI: 10.26868/25222708.2019.210945.

[49] Arie Taal, Laure Itard, and Wim Zeiler. "A reference architecture for the integration of automated energy performance fault diagnosis into HVAC systems". In: *Energy and Buildings* 179 (2018), pp. 144–155. ISSN: 03787788. DOI: 10.1016/j.enbuild.2018.08.031. URL: https://doi.org/10.1016/j.enbuild.2018.08.031.

[50] Anand Thamban. *Fault detection and diagnosis of the low ΔT syndrome in cooling coils of chilled water systems*. Tech. rep. 2022.

[51] Anand Thamban, Alet van den Brink, Shalika Walker, and Wim Zeiler. "Detection of the low ΔT syndrome using machine learning models". In: *CLIMA 2022 conference* (May 2022). DOI: 10.34641/CLIMA.2022.121. URL: https://proceedings.open.tudelft.nl/clima2022/article/view/121.

[52] The European Parliament and the Council of the EU. "Directive 2010/31/EU of the European parliament and of the council of 19 May 2010 on the energy performance of buildings (recast)". In: *Official Journal of the European Communities (2010)* L153/13-35 (2010).

[53] Khin Me Me Thein. "Apache kafka: Next generation distributed messaging system". In: *International Journal of Scientific Engineering and Technology Research* 3.47 (2014), pp. 9478–9483.

[54] Deepak Vohra. "Apache Avro". In: *Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools*. Berkeley, CA: Apress, 2016, pp. 303–323. ISBN: 978-1-4842-2199-0. DOI: 10.1007/978-1-4842-2199-0{_}7. URL: https://doi.org/10.1007/978-1-4842-2199-0_7.

[55] Yalan Wang, Zhiwei Wang, Suowei He, and Zhanwei Wang. "A practical chiller fault diagnosis method based on discrete Bayesian network". In: *International Journal of Refrigeration* 102 (2019), pp. 159–167. ISSN: 01407007. DOI: 10.1016/j.ijrefrig.2019.03.008. URL: https://doi.org/10.1016/j.ijrefrig.2019.03.008.

[56] Z Wang, Chujie Lu, A Taal, G Srinivasan, K Mohammed, Arjen Meijer, and Laure Itard. "Bayesian network-based fault detection and diagnosis of heating components in heat recovery ventilation". In: *research.tudelft.nlZ Wang, C Lu, A Taal, S Gopalan, K Mohammed, A Meijer, L ItardRoomvent 2024, 2024•research.tudelft.nl*. 2024. URL: `https://research.tudelft.nl/en/publications/bayesian-network-based-fault-detection-and-diagnosis-of-heating-c`.

[57] Ziao Wang, Arjen Meijer, and Laure Itard. "4S3F Diagnostic Bayesian Network method : discussion about application and technical design". In: *CLIMA 2022 Conference* (2022), pp. 1–8. DOI: `https://doi.org/10.34641/clima.2022.82`.

[58] Fu Xiao, Yang Zhao, Jin Wen, and Shengwei Wang. "Bayesian network based FDD strategy for variable air volume terminals". In: *Automation in Construction* 41 (2014), pp. 106–118.

[59] Yuebin Yu, Denchai Woradechjumroen, and Daihong Yu. "A review of fault detection and diagnosis methodologies on air-handling units". In: *Energy and Buildings* 82 (2014), pp. 550–562. DOI: `10.1016/j.enbuild.2014.06.042`. URL: `http://dx.doi.org/10.1016/j.enbuild.2014.06.042`.

[60] Yang Zhao, Tingting Li, Xuejun Zhang, and Chaobo Zhang. "Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future". In: *Renewable and Sustainable Energy Reviews* 109.April (2019), pp. 85–101. ISSN: 18790690. DOI: `10.1016/j.rser.2019.04.021`. URL: `https://doi.org/10.1016/j.rser.2019.04.021`.

[61] Yang Zhao, Jin Wen, and Shengwei Wang. "Diagnostic Bayesian networks for diagnosing air handling units faults - Part II: Faults in coils and sensors". In: *Applied Thermal Engineering* 90 (2015), pp. 145–157. ISSN: 13594311. DOI: `10.1016/j.applthermaleng.2015.07.001`.

[62] Yang Zhao, Jin Wen, Fu Xiao, Xuebin Yang, and Shengwei Wang. "Diagnostic Bayesian networks for diagnosing air handling units faults – part I: Faults in dampers, fans, filters and sensors". In: *Applied Thermal Engineering* 111 (2017), pp. 1272–1286. ISSN: 13594311. DOI: `10.1016/j.applthermaleng.2015.09.121`.

A

Fault experiments

Table A.1: Conducted experiments dates and times.

Date	Fault	Start time	End time
03/23/2024	Supply fan control set to 30%	8:21	11:14
	Supply fan control set to 70%	11:14	14:32
	ERW control set to 0%	14:32	17:00
03/24/2024	Supply pressure setpoint set to 235 Pa	11:05	13:58
	Supply pressure setpoint set to 135 Pa	13:58	17:00
	Supply pressure sensor set to 130 Pa	17:00	20:00
03/29/2024	HCV control set to 0%	9:25	11:24
	HCV control set to 30%	11:24	14:00
	HCV control set to 100%	14:00	17:12
03/30/2024	Supply pressure sensor set to 200 Pa	9:53	10:56
	Temperature setpoint set to 23 °C	11:22	14:35
	Temperature setpoint set to 17 °C	14:35	17:59
	Supply temperature sensor set to 17 °C	18:10	20:00

B

Revised DBN

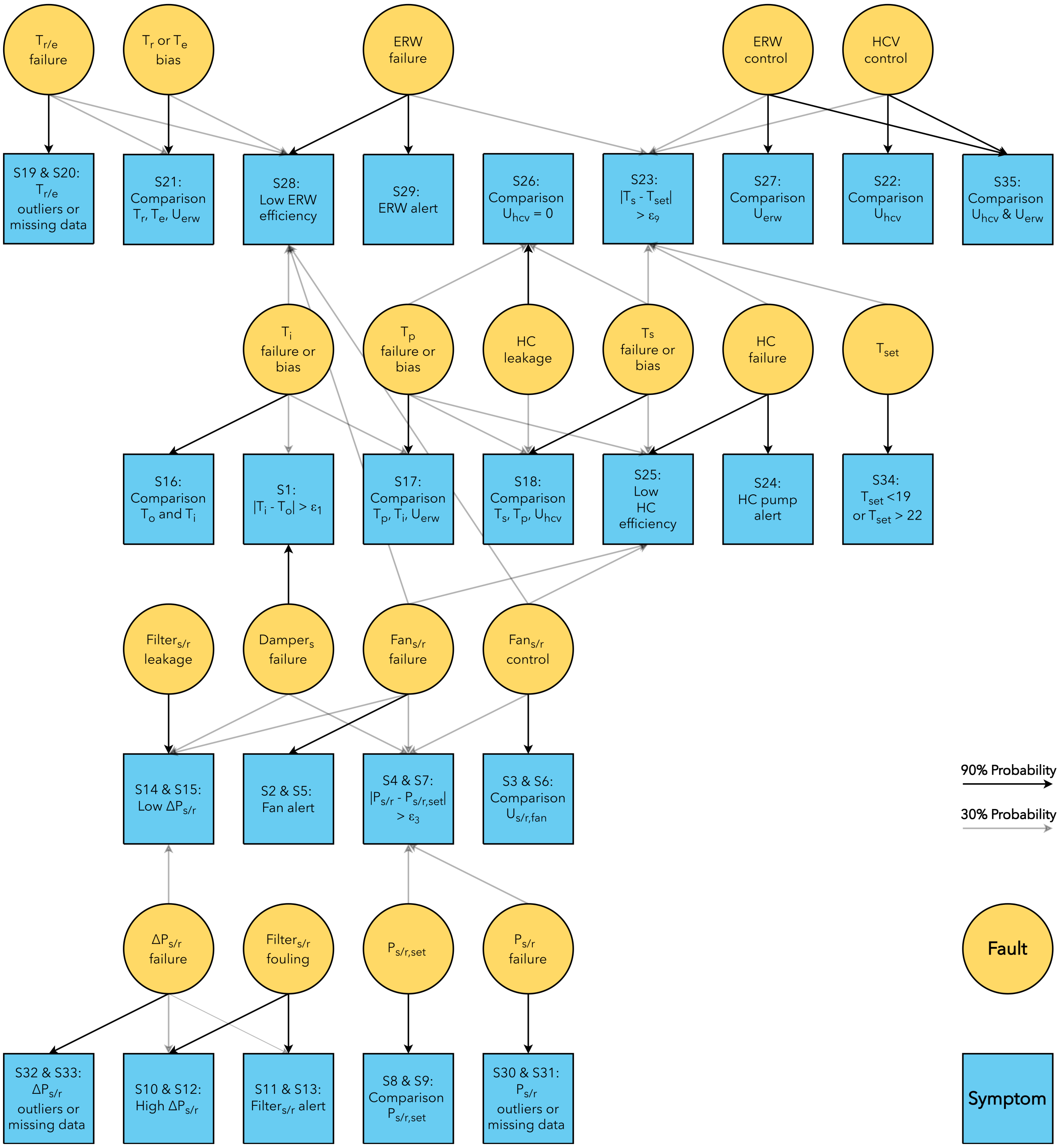


Figure B.1: Revised DBN with symptom S35.

	Symptom	Rule
S_1	Comparison T_i and T_o	$\lvert T_i - T_o \rvert > \varepsilon_1$
S_2	Supply fan alert is present	$A_{s,fan} = 1$
S_3	Comparison $U_{s,fan}$, $p_{s,set}$ and p_s	$U_{s,fan} < 0\%$ or $U_{s,fan} > 100\%$ or $(U_{s,fan} < 100\%$ and $p_s < p_{s,set} - \varepsilon_2)$ or $(U_{s,fan} > 0\%$ and $p_s > p_{s,set} + \varepsilon_2)$
S_4	Difference $p_{s,set}$ and p_s	$\lvert p_s - p_{s,set} \rvert > \varepsilon_3$
S_5	Return fan alert is present	$A_{r,fan} = 1$
S_6	Comparison $U_{r,fan}$, $p_{r,set}$ and p_r	$U_{r,fan} < 0\%$ or $U_{r,fan} > 100\%$ or $(U_{r,fan} < 100\%$ and $p_r < p_{r,set} - \varepsilon_2)$ or $(U_{r,fan} > 0\%$ and $p_r > p_{r,set} + \varepsilon_2)$
S_7	Difference $p_{r,set}$ and p_r	$\lvert p_r - p_{r,set} \rvert > \varepsilon_3$
S_8	Comparison $p_{s,set}$	$\lvert p_{s,set} - 185 \rvert > \varepsilon_4$
S_9	Comparison $p_{r,set}$	$\lvert p_{r,set} - 200 \rvert > \varepsilon_4$
S_{10}	High Δp_s	$\Delta p_s > \varepsilon_{5s}$
S_{11}	Supply filter alert present	$A_{s,filter} = 1$
S_{12}	High Δp_r	$\Delta p_r > \varepsilon_{5r}$
S_{13}	Return filter alert present	$A_{r,filter} = 1$
S_{14}	Low Δp_s	$U_{s,fan} \geq 5$ and $\Delta p_s < \varepsilon_6$
S_{15}	Low Δp_r	$U_{r,fan} \geq 5$ and $\Delta p_r < \varepsilon_6$
S_{16}	Difference T_i and T_o	$\lvert T_o - T_i \rvert > \varepsilon_7$ or $T_i = NaN$ or $T_i < \varepsilon_{14}$ or $T_i > \varepsilon_{15}$
S_{17}	Difference T_p and T_i with ERW off	$(\lvert T_p - T_i \rvert > \varepsilon_7$ and $U_{erw} = 0\%)$ or $T_p = NaN$ or $T_p < \varepsilon_{14}$ or $T_p > \varepsilon_{15}$
S_{18}	Difference T_s and T_p with HCV & CCV closed	$(\lvert T_p - T_s \rvert > \varepsilon_7$ and $U_{hcv} = 0\%$ and $U_{ccv} = 0\%)$ or $T_s = NaN$ or $T_s < \varepsilon_{14}$ or $T_s > \varepsilon_{15}$
S_{19}	T_r outlier or missing data	$T_r = NaN$ or $T_r < \varepsilon_{14}$ or $T_r > \varepsilon_{15}$
S_{20}	T_e outlier or missing data	$T_e = NaN$ or $T_e < \varepsilon_{14}$ or $T_e > \varepsilon_{15}$
S_{21}	Difference T_r and T_e with ERW off	$\lvert T_r - T_e \rvert > \varepsilon_7$ and $U_{erw} = 0\%$
S_{22}	Comparison U_{hcv}, T_s and T_{set}	$(U_{hcv} > 0\%$ and $T_s > T_{set} + \varepsilon_8)$ or $(U_{hcv} < 100\%$ and $T_s < T_{set} - \varepsilon_8)$ or $U_{hcv} < 0\%$ or $U_{hcv} > 100\%$
S_{23}	Difference T_s and T_{set}	$\lvert T_s - T_{set} \rvert > \varepsilon_9$
S_{24}	Heating coil pump error present	$A_{hc} = 1$
S_{25}	Low heating coil efficiency	$(T_s - T_p) < \varepsilon_{10} * \frac{U_{hcv}}{100}$ and $U_{hcv} > 0\%$
S_{26}	Comparison T_p and T_s with HCV closed	$U_{hcv} = 0\%$ and $T_s > T_p + \varepsilon_{11}$
S_{27}	Comparison U_{erw}, T_s and T_{set}	$(U_{erw} < 100\%$ and $T_s < T_{set} - \varepsilon_{12}$ and $T_r > T_o)$ or $(U_{erw} > 0\%$ and $T_r < T_o)$ or $(U_{erw} > 0\%$ and $T_s > T_{set} + \varepsilon_{12})$ or $U_{erw} < 0\%$ or $U_{erw} > 100\%$
S_{28}	Low ERW efficiency	$U_{erw} > 0\%$ and $(\frac{T_r - T_e}{T_r - T_i} < \varepsilon_{13} * \frac{U_{erw}}{100})$
S_{29}	ERW alert present	$A_{erw} = 1$
S_{30}	p_s outlier or missing data	$p_s = NaN$ or $(p_s < \varepsilon_{16})$ or $(p_s > \varepsilon_{17})$
S_{31}	p_r outlier or missing data	$p_r = NaN$ or $(p_r < \varepsilon_{16})$ or $(p_r > \varepsilon_{17})$
S_{32}	Δp_s outlier or missing data	$\Delta p_s = NaN$ or $(\Delta p_s < \varepsilon_{16})$ or $(\Delta p_s > \varepsilon_{17})$
S_{33}	Δp_r outlier or missing data	$\Delta p_r = NaN$ or $(\Delta p_r < \varepsilon_{16})$ or $(\Delta p_r > \varepsilon_{17})$
S_{34}	High or low T_{set}	$T_{set} < \varepsilon_{18}$ or $T_{set} > \varepsilon_{19})$
S_{35}	Comparison U_{hcv} and U_{erw}	$U_{hcv} > 0\%$ and $U_{erw} < 100\%$

Table B.1: Symptoms detection rules for the DBN including symptom S35 and revised rules for symptoms S22 and S27.

C
Python packages

Table C.1: Python packages used.

Package	version
confluent-kafka	2.4.0
influxdb-client	1.43.0
numpy	1.26.4
pandas	2.2.1
pgmpy	0.1.25

D
Example Kafka message


```json
"system_manufacturer": "",
"building": "28",
"brick_system_class": "HVAC",
"timestamp": 1719224913464,
"equipments": [
    {
      "equipment_name": "AHU␣1",
      "brick_equipment_class": "AHU-Pressure_Sensor",
      "equipment_description": "Pressure␣transmitter",
      "equipment_reference_id": "",
      "equipment_NLSfB_element_class": "",
      "equipment_NLSfB_element_naam": "",
      "location_wing": "D",
      "location_floor": "",
      "location_room_number": "",
      "location_room_name": "",
      "location_room_description": "",
      "location_room_class": "",
      "location_zone": "",
      "location_zone_description": "",
      "measurements": [
        {
          "measurement_reference_id": "",
          "measurement_id_name": "(28)␣201.PT-02M--",
          "measurement_name": "Pressure␣transmitter-Measurement-None",
          "brick_point_class": "Sensor",
          "unit": "pascals",
          "value": 200.441833,
          "measurement_description": ""
        }
      ]
    }
],
"portfolio": "",
"system_manufacturer_description": "",
"portfolio_description": "",
"building_description": "",
"brick_system_class_description": ""
```


Listing D.1: Example case study Kafka message with sensor data, deserialised to JSON format.