TOR VERGATA
UNIVERSITY OF ROME

PhD in
Industrial Engineering

Cycle
XXXIV

A Predictive Maintenance Model for Heterogeneous Industrial Refrigeration Systems

Ron van de Sand, M. Eng.

A.Y. 2020/2021

Tutor: Prof. Dr. Sandra Corasaniti

Co-Tutor: Prof. Dr.-Ing. Jörg Reiff-Stephan

Coordinator: Prof. Dr. Marco Marinelli

Abstract

The automatic assessment of the degradation state of industrial refrigeration systems is becoming increasingly important and constitutes a key-role within predictive maintenance approaches. Lately, data-driven methods especially became the focus of research in this respect. As they only rely on historical data in the development phase, they offer great advantages in terms of flexibility and generalisability by circumventing the need for specific domain knowledge. While most scientific contributions employ methods emerging from the field of machine learning (ML), only very few consider their applicability amongst different heterogeneous systems. In fact, the majority of existing contributions in this field solely apply supervised ML models, which assume the availability of labelled fault data for each system respectively. However, this places restrictions on the overall applicability, as data labelling is mostly conducted by humans and therefore constitutes a non-negligible cost and time factor. Moreover, such methods assume that all considered fault types occurred in the past, a condition that may not always be guaranteed to be satisfied.

Therefore, this dissertation proposes a predictive maintenance model for industrial refrigeration systems by especially addressing its transferability onto different but related heterogeneous systems. In particular, it aims at solving a sub-problem known as *condition-based maintenance* (CBM) to automatically assess the system's state of degradation. To this end, the model does not only estimate how far a possible malfunction has progressed, but also determines the fault type being present. As will be described in greater detail throughout this dissertation, the proposed model also utilises techniques from the field of ML but rather bypasses the strict assumptions accompanying supervised ML. Accordingly, it assumes the data of the target system to be primarily unlabelled while a few labelled samples are expected to be retrievable from the fault-free operational state, which can be obtained at low cost. Yet, to enable the model's intended functionality, it additionally employs data from only one fully labelled source dataset and, thus, allows the benefits of data-driven approaches towards predictive maintenance to be further exploited.

After the introduction, the dissertation at hand introduces the related concepts as well as the terms and definitions and delimits this work from other fields of research. Furthermore, the scope of application is further introduced and the latest scientific work is presented. This is then followed by the explanation of the open research gap, from which the research questions are derived. The third chapter deals with the main principles of the model, including the mathematical notations and the individual concepts. It furthermore delivers an overview about the variety of problems arising in this context and presents the associated solutions from a theoretical point of view. Subsequently, the data acquisition phase is described, addressing both the data collection procedure and the outcome of the test cases. In addition, the considered fault characteristics are presented and compared with the ones obtained from the related publicly available dataset. In essence, both datasets form the basis for the model validation, as discussed in the following chapter. This chapter then further comprises the results obtained from the model, which are compared with the ones retrieved from several baseline models derived from the literature. This work then closes with a summary and the conclusions drawn from the model results. Lastly, an outlook of the presented dissertation is provided.

Acknowledgements

This dissertation was written during my time as a research engineer at the Technical University of Applied Sciences Wildau in the IC3-Smart Production research group, of which I am proud to be a member and even prouder to have taken on the role of the deputy head during the last years of my research. Special thanks are due to Prof. Dr.-Ing. Jörg Reiff-Stephan, who has guided me within the scope of my research in countless meetings and discussions over the years and helped to shape the concept of this work. My gratitude is also owned to the University of Rome "Tor Vergata", in particular to Prof. Dr. Sandra Corasaniti, for accepting me into the PhD program and for supporting my work with any questions or problems I had.

Furthermore, I would like to thank the pakt GmbH, especially Sebastian Querner and Robert Schwinkendorf, for providing the chiller and for their support during the test rig development as well as during the troubleshooting in the commissioning phase. Not to forget my colleagues and the students, especially Constantin Falk, Willy Palme and Andreas Krispin, who offered their help during the whole experimental phase and without whom the multitude of tasks would not have been possible to solve.

I also want to thank my family, in particular my parents Katrin and Dieter van de Sand, and friends who I didn't get to see very often in the past years. Last but not least, I'd like to express my greatest appreciation and gratitude to my partner Laura Flöther, who always supported me with her patience and lots of good advice during endless working days.

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

— Alan Turing

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

ACC	Accuracy
ANN	Artificial Neural Network
ARX	Auto-Regressive Model with Exogenous Variables
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
AUC	Area Under the Curve - Performance Metric from the ROC
BSVM	Biased Support Vector Machine
CBM	Condition Based Maintenance
CDFD	Cross Domain Fault Diagnosis
CM	Condition Monitoring
CORAL	Correlation Alignment
CPV	Cumulative Percent Value
FDD	Fault Detection and Diagnosis
FDI	Fault Detection and Isolation
F-MOD	Modified F-Score Metric
FN	Number of False Negatives
FNR	False Negative Rate
FP	Number of False Positives
FPR	False Positive Rate
HMI	Human Machine Interface
LDA	Linear Discriminant Analysis
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MLR	Multiple Linear Regression
MMD	Maximum Mean Discrepancy
NC	Non-Condensables
OCC	One Class Classification
OC-SVM	One Class - Support Vector Machine
OPC-UA	Open Platform Communication - Unified Architecture
OSA-CBM	Open System Architecture for Condition-Based Maintenance
PC	Principal Component
PCA	Principal Component Analysis
PCS	Principal Component Subspace
PHM	Prognostic Health Management
PLC	Programmable Logic Controller

PM	Predictive Maintenance
PMS	Power Monitoring System
PU	Positive-Unlabelled
RBF	Radial Basis Function
RC	Residual Component
RCS	Residual Component Subspace
RKHS	Reproducing Kernel Hilbert Space
RL	Refrigerant Leakage
ROC	Receiver operating characteristic
RSA	Residual subspace alignment
RUL	Remaining Useful Life
RVC	Reduced Volume Flow at Condenser
RVE	Reduced Volume Flow at Evaporator
SA	Subspace Alignment
SCL	Structural Correspondence Learning
SL	Severity Level
SNR	Signal-to-Noise Ratio
SPE	Squared Prediction Error
SV	Support Vector
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TCA	Transfer Component Analysis
TN	Number of True Negatives
TNR	True Negative Rate
TP	Number of True Positives
TPR	True Positive Rate
VFD	Variable-Frequency Drive

Symbols

Latin symbols

A_A, B_A, C_A	Antoine coefficients
b	Bias term
C, C^+, C^-	Regularisation parameter
CPV	Cumulative percent value of the explained variance
$CPV_{k_{pc}}$	Minimum CPV to be discarded, a threshold parameter
c_p	Specific heat capacity
$\mathcal{D}$	Domain
d	Set of disturbances
$f(\cdot)$	Designation of a function
fti	Fault type
$h(\cdot)$	Classifier
$h(\cdot)^*$	Decision function
h_e	Specific enthalpy
Δh_e	Specific enthalpy difference
hi	Health index
I_{comp}	Instantaneous motor current (compressor)
K	Kernel matrix
k	Number of dimensions
$\mathbf{k}$	Number of folds in k-fold cross-validation
$k(\cdot)$	Kernel function
k_{pc}	Number of principal components
k_{tca}	Number of transfer components (TCA)
m	Number of fault classes
$\dot{m}_C$	Cooling water mass flow
$\dot{m}_E$	Chilled water mass flow
n	Number of observations
n_{comp}, n^*_{comp}	Compressor rotational speed and its set-point variable
$P, \hat{P}, \tilde{P}$	PCA loading matrix, its PCS and its RCS part
$\mathcal{P}(X), \mathcal{P}(Y\|X)$	Marginal and conditional probability distribution
P_{comp}	Instantaneous motor power (compressor)
pd	Polynomial degree
p_{rc}	Refrigerant pressure at compressor outlet
p_{re}	Refrigerant pressure at compressor inlet

$Q_C, \dot{Q}_C$	Condenser heat and heat flow
$Q_E, \dot{Q}_E$	Evaporator heat and heat flow
R^2	Coefficient of determination
r	Recall
$\mathcal{T}$	Learning task
$T_{amb,i}$	Test rig ambient temperature (inside)
$T_{amb,o}$	Test rig ambient temperature (outside)
ΔT_C	Condenser water temperature difference
ΔT_{ca}	Condenser approach temperature
T_{ci}, T_{ci}^*	Water temperature at condenser inlet and its set-point variable
T_{co}	Water temperature at condenser outlet
T_{dis}	Refrigerant discharge temperature
ΔT_E	Evaporator water temperature difference
ΔT_{ea}	Evaporator approach temperature
T_{ei}, T_{ei}^*	Water temperature at evaporator inlet and its set-point variable
T_{eo}	Water temperature at evaporator outlet
t_{min}	Minimum steady-state time
T_{oil}	Oil feeding temperature
T_{rc}	Refrigerant condensing temperature
T_{re}	Refrigerant evaporation temperature
$\Delta T_{sh,dis}$	Refrigerant discharge superheat temperature
$\Delta T_{sh,suc}$	Refrigerant suction superheat temperature
T_{suc}	Refrigerant suction temperature
u	Set of input variables
$\dot{V}_C$	Cooling water volume flow rate
$\dot{V}_E$	Chilled water volume flow rate
w	Vector orthogonal to the hyperplane (SVM)
ws	Window size
w_t	Induced work
$X, \tilde{X}$	Input matrix and its transformation to the RCS
$\mathcal{X}$	Feature space
$\grave{x}_i$	Filtered observation recorded from the monitored system
$x_i, \tilde{x}_i$	Observation and its transformation to the RCS
Y	Collection of labels
$\mathcal{Y}$	Label space
y_i	Label

y_v	Approximated variable from the regression model
Z	Matrix derived from a data subset
$z_i, \tilde{z}_i$	Observation from subset and its transformation to the RCS

Indices

$bsvm$	Biased support vector machine
fd	Fault detection
fi	Fault isolation
fs	Data associated to faults of the source domain
i, j, k	Various indices
n, ns, nt	Data associated to the normal operation conditions
$ocsvm$	One-class support vector machine
s	Source domain
t	Target domain
u	Data associated to unknown observations in the deployment phase

Greek symbols

α_i, β_i	Lagrangian coefficients
β_i^*	Regression coefficients
γ	Width parameter of the RBF kernel
δ_e	Deviation from benchmark dataset
δ_{thr}	Steady-state threshold
ε	Coefficient of performance
θ	Fraction of normal labelled data
Λ	Diagonal matrix containing eigenvalues from eigendecomposition
λ	Regularisation parameter (CORAL)
μ	Mean vector
ξ	Slack variable
ρ	Bias term of the OC-SVM decision function
ρ_w	Water density
Σ	Covariance matrix
σ	Standard deviation
σ_{dev}	Standard deviation of time derivatives
$\Phi(\cdot)$	Mapping function to feature space

1 Introduction

This dissertation is about a novel approach towards predictive maintenance for industrial refrigeration systems. The proposed model aims to bridge the gap between research and practical implementation by being specifically designed for its transferability between different but related heterogeneous systems. In this first chapter, the main terms arising in this context are introduced and the scope of this work is further explained. Furthermore, the related research works are discussed and the open research gap is derived.

1.1 Motivation

The field of *predictive maintenance* (PM) has attracted researchers for many years and is known to play a key-role in the transformation process from conventional manufacturing strategies towards highly efficient value chains in the course of 'Industry 4.0'. By combining new emerging technologies, such as machine learning (ML), artificial intelligence, big data or the internet of things [1], PM is changing the face of production activities [2] as it enables automated real time degradation assessment of technical equipment. It has, thus, developed to a prominent strategy to minimise machinery downtime and associated costs [3]. Based on actual machine data, this methodology tries to reduce unnecessary maintenance actions [4] and to predict unplanned downtimes [5, p. 37] or efficiency losses [6] of machine components. In contrast to other maintenance approaches, it enables to schedule maintenance actions depending on the actual machine condition [7, pp. 7-9] rather than on predetermined intervals or after machine failure, which is often referred to as maintenance on demand. This leads to increased system availability from the perspective of the overall equipment effectiveness [8, p. 37] and, thus, may lower operation and maintenance expenditures. Furthermore, it is known to improve the productivity and product quality of a production plant as well as to increase operator safety [9, pp. 61-72]. As an actual alternative to the existing state-of-the-art maintenance types, PM has recently evolved into a basis for new business models.

In the survey "Predictive Maintenance – Market Report 2019 - 2024" [10], the authors predicted a worldwide rise in the predictive maintenance market from US$ 3.3 billion in 2018 to US$ 23.5 billion in 2024, which indeed illustrates the importance of the issue. This is moreover underlined by the fact that maintenance tasks are increasingly conducted as services through service providers or the manufacturers themselves [11, p. 19]. For example, the Federal Statistical Office of Germany has identified around thirteen thousand companies in this sector with a total revenue of 3.6 billion Euro for the year 2017, of which around two thousand companies are active in the field of data processing and telecommunications equipment [12]. In the same year, the German Engineering Federation concluded that 81 % of the companies within the German engineering sector considered PM to be an important industry trend and 40 % even considered it to be a differentiator and success factor for future businesses [13]. With data becoming more accessible and ubiquitous [14], the use of data-driven PM methods has been especially promoted in recent years and has become a well established solution [15] for the automatic degradation assessment of technical equipment. As such approaches observe the underlying pattern of machine data instead of relying on a mathematical formulation of the physical system [16], they require no domain knowledge [17, p. 28] and are therefore particularly suitable for technical systems with high complexity. On the contrary, however, data-driven PM methods commonly utilise training datasets with labelled observations that may suffer

from the lack of faulty samples [18], as faults are rare events [6]. In addition, data labelling is still performed by humans and therefore constitutes a considerable cost factor [19], which inhibits the broad application of such approaches.

Particularly in the field of heating, ventilation, air-conditioning and refrigeration systems (HVAC&R), PM offers great potential for energy and cost savings. In the European Union alone, HVAC&R applications account for about half of the energy demand [20] and even more for countries with higher temperatures and humidity. In Abu Dhabi for instance, 61 % of the total annual energy consumption can be ascribed to such applications, of which 47 % is accounted for by chillers [21]. As described in [22], these figures will most likely increase in the upcoming years due to climate change, but can also be affected by factors beyond that, such as economic and population growth, or social and cultural trends. This, in conjunction with the fact that refrigeration systems can lose as much as 30 % efficiency due to performance degradation whilst appearing fully functional [23], shows the potential for reducing energy waste [24] through proper maintenance strategies. Especially in the industrial refrigeration sector, PM can add significant value in terms of process reliability. One particularly important branch is the food industry, where chiller breakdowns may lead to the obstruction of the production process causing reduced product quality or even product loss [25]. But also other branches, such as the chemical or pharmaceutical industry, rely on properly maintained refrigeration systems and can therefore benefit from the advantages of PM.

Although data-driven approaches offer many advantages, their broad application in the industrial refrigeration sector is not yet well advanced. This is mainly due to the fact that the industry is dominated by custom-made chillers built on-site and thus vary in their system design as well as their operating characteristics [26, p. 464]. As a consequence, machine data acquired from various domains are subject to different distributions leading to inconsistent classification behaviour if one PM model is applied across different domains. Common state-of-the-art models are therefore often trained independently for each system using labelled normal and fault data samples, resulting in high costs in their development.

One solution to this problem is represented through the application of transfer learning or, as will be introduced later, *domain adaptation* approaches as has already been successfully applied in other engineering sectors, such as robotics [27] or rolling bearing diagnostics [28]. Such transferable models are of great value as they allow the benefits of data-driven approaches to be further exploited. By embedding prior-knowledge [29] from a fully labelled source domain, such models may adapt to a target domain where few or no labelled observations are available. A major problem, however, is the insufficient number of available datasets of different chiller types [30], which may be important [6] for promoting transferable data-driven diagnostic models and accelerating progress in this research area. Even though many PM approaches for assessing a chiller's degradation state by using associated labelled data are well described across the literature, practices that avoid costly data labelling are still lacking and remain a challenge in this field.

1.2 Scope

Despite the advantages of data-driven PM, the lack of sufficiently labelled datasets is still a major obstacle [4], hindering its widespread application. Yet, this statement does not necessarily hold for the availability of labelled data stemming from the normal, or fault-free, operation condition as these can be obtained during the commissioning phase

or directly after maintenance measures have been performed. Similarly, unlabelled data are often readily available [31, p. 610] in great numbers because many system operators already collect data as part of their control strategies [32] for optimisation, documentation or traceability purposes.

As will be pointed out in the following sections, the main scope of this dissertation is on a certain subsystem of the refrigeration cycle, namely the chiller, as it is critical with regard to malfunctions and energy losses [33]. Therefore, based on the aforementioned aspects, a novel data-driven model to automatically assess a chiller's degradation state is proposed in this dissertation. More specifically, the model enables the detection of faults and supports the analysis of their root cause by providing comprehensive fault diagnosis capabilities. This process involves a multi-level approach known as *condition-based maintenance* (CBM) [34, p. 1], often used as a synonym for PM. The difference between both terms as well as the distinction to related research areas will be discussed in more detail in the following section.

In general, the proposed model deals with a common problem originating from data-driven methods, namely the absence of labels. In fact, the model aims at utilising only a minor fraction of available labelled data stemming from the normal chiller operation as well as a greater number of unlabelled data in the training phase. This is achieved by exploiting methodologies from the field of one-class classification (OCC) [35] and domain adaptation [36], both of which are problems in ML. While the former is used for anomaly detection tasks, the latter is used for estimating the fault type by embedding prior-knowledge from another fully labelled chiller dataset, the source dataset.

The main goal of the proposed model is more ambitious than requiring only fewer labelled data samples of the target system in the training phase. It rather aims at exploiting primarily unlabelled data with a small number of labelled observations associated with the normal operating condition and, thus, avoids costly and time-consuming data labelling tasks. This dissertation therefore describes the principles of the model, the underlying ideas, and validates its general functionality. To this end, two datasets are employed, one of which was collected within the scope of this work.

1.3 Outline

This dissertation is divided into six chapters, of which this introduction is the first. Chapter 2 introduces the terms related to industrial maintenance but also to ML. Another important aspect of this chapter concerns the delimitation of this work from other fields of research, whereby the categorisation of the sub-fields within the context of PM is clarified. Yet, the most important contribution of this chapter lies in the derivation of the research gap and research questions arising from it. The third chapter describes the problem at hand from a theoretical perspective and introduces the mathematical notations used throughout this work. Moreover, the algorithmic principles of the proposed model are described, starting from an abstract description to a detailed procedure. The subsequent chapter then deals with the presentation of the test rig and the experiments conducted to collect the dataset, which, in turn, is used to validate the model. Moreover, the investigated fault patterns are discussed and compared with the results obtained from other studies. Chapter 5 demonstrates the overall functionality of the model with a particular focus on its classification performance, for which a comparison is made with common state-of-the-art algorithms. Finally, the results are summarised and an outlook is given in the concluding chapter.

2 State of the Art

In the following, the topic of predictive maintenance is described in more detail and its individual components are explained. Furthermore, an overview of the current state of the art with regard to industrial refrigeration systems is provided and the main terms are introduced and differentiated from each other. Most importantly, the relationship between PM and condition based maintenance is presented. This is followed by the introduction of this technology to the field of industrial refrigeration, whereby typical fault types are presented and current challenges are discussed. The final part covers the related works in this research field, from which the open research gap is derived in the concluding section.

2.1 Industrial Maintenance

The term *maintenance* is i.a. defined within the DIN EN 13306 as the "*combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function*" [37, p. 8]. This does not only include active measures such as repair or refurbishment, but also comprises observation and analysis of a technical asset's state. It should be noted that the latter is the main subject of the present work, focusing on the automated analysis procedure.

2.1.1 Objectives

In the last decades, industrial maintenance has garnered increased attention, as more and more production companies see it as a factor to generate added value for their business processes [38, p. 17]. As stated in [39, p. 10], its scope has further evolved from solely ensuring the availability of machinery. Nowadays, industrial maintenance stretches well beyond that, as it has become an aspect of development and competitiveness. Thus, it is often associated with the reduction of production costs or the increase of product quality. In [37, p. 9], the maintenance objectives are summarised as follows:

- Ensure availability of machinery
- Cost reduction
- Increase useful life
- Product quality
- Environment preservation
- Asset value preservation
- Safety

As outlined in [40, p. 2], maintenance can even be considered a value-adding factor, as it is the preliminary work for the actual value-added process. Although maintenance itself is typically not considered part of the value chain, it offers enormous potential for its improvement [41]. Stephens [42, p. 6] distinguished between primary and secondary goals of maintenance. Primary goals include early and appropriate response to equipment failures, development of critical situation detection procedures, or improvement of system efficiency. Secondary goals, on the other hand, are not directly related to the production process but may include measures to improve plant protection and security or to reduce pollution and noise.

In general, planning maintenance procedures for a certain asset is a trade-off between system reliability and maintenance costs [44]. This is described in [43, p. 25] and [42, p. 3-7], among others, where maintenance-related costs and the degree of preventive measures

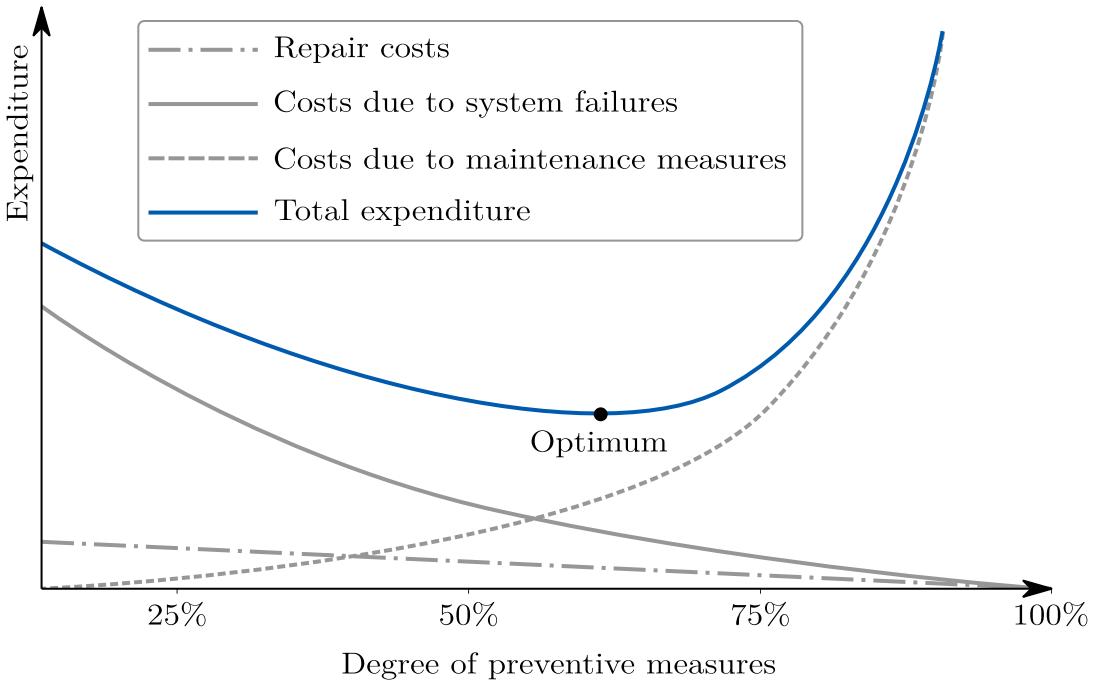

Figure 2.1: The maintenance optimisation problem as a trade-off between system reliability and maintenance costs according to [43, p. 24].

are compared in the context of improving system reliability. As shown in Figure 2.1, both repair costs and costs caused by unscheduled system downtimes decrease with a rising degree of preventive measures. This contrasts with a disproportionate increase in the costs incurred by the preventive measures to satisfy higher system reliability requirements [43, pp. 22-24]. By adding up these costs, it becomes apparent that both a too low as well as a too high degree of preventive measures cause excessive costs and that the costs for ensuring a 100 % preventive maintenance strategy even tend towards infinity. Besides, the technical feasibility of such a measure would be questionable. Consequently, the definition of a maintenance strategy for an asset can be considered as an optimisation problem, with the objective of achieving the highest reliability at minimum cost. Therefore, selecting a suitable strategy is of great importance to achieve the aforementioned objectives.

2.1.2 Strategies

Industrial maintenance has progressively become a major concern for manufacturing companies over the years and has also changed with the appearance of new requirements and technologies. As a result, various concepts have emerged since the first industrial revolution, as shown in Figure 2.2. While the 19$^{\text{th}}$ century was primarily characterised by reactive maintenance measures, preventive strategies were increasingly applied from the middle of the 20$^{\text{th}}$ century. From the 1970s onwards, minimising operating costs over the entire product life cycle became more important [45], which led to initial attempts to plan maintenance measures based on the actual asset condition. Furthermore, with increasing numbers of real-time sensor data being available as well as the introduction of computers into the production processes, maintenance strategies became more advanced.

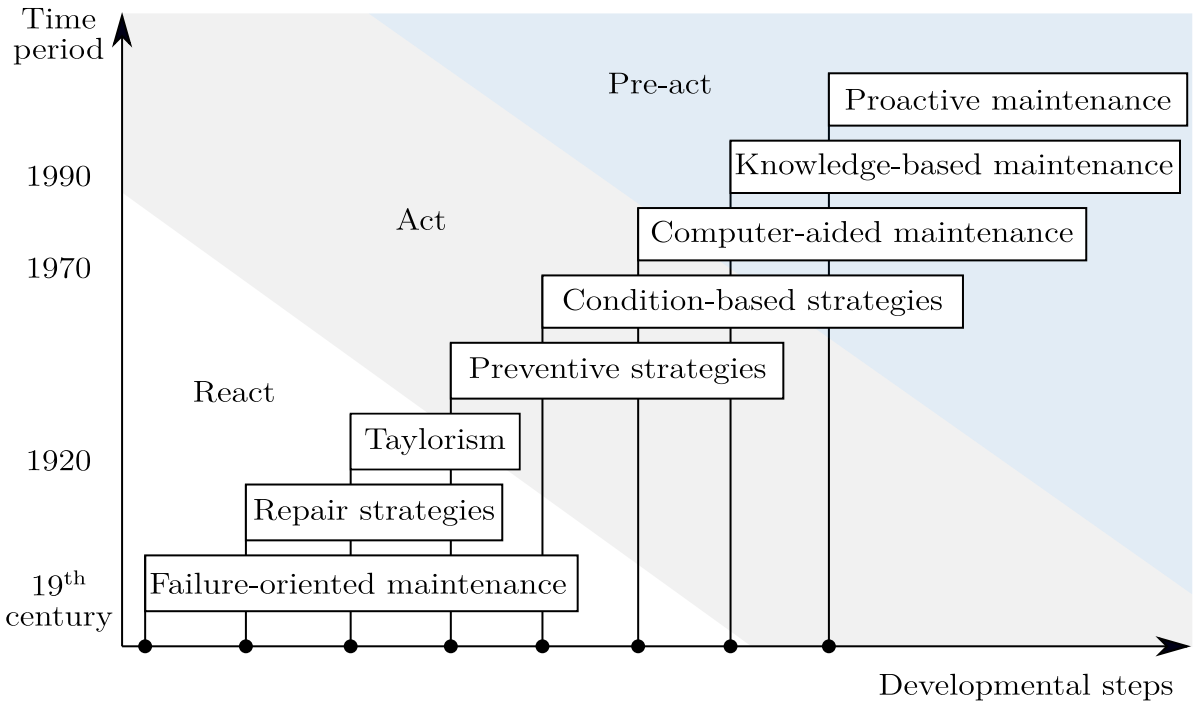

Figure 2.2: Developmental stages of maintenance strategies derived from [40, p. 2].

This has led to a paradigm shift towards proactive planning of maintenance tasks using new mathematical methods, which is still a field of ongoing development [46, p. 22].

This does not necessarily mean that nowadays all maintenance measures are purely proactive in nature. In fact, most maintenance actions are still reactive [47, p. 9] and consequently repair measures are accorded a high degree of significance for the maintenance staff. As described in the previous section, the trade-off between reliability and the associated costs determines the extent of the measures required to maintain the technical equipment. The individual concept comprising these measures is usually referred to as the *maintenance strategy*, which is to be defined by the responsible management [37]. Yet, this should not be confused with maintenance types, as maintenance strategies rather describe how such types are combined to achieve certain technical benefits [38, p. 373].

Mikat [48, pp. 15-17] describes three maintenance strategies that can be distinguished; *run-to-failure-maintenance*, *on-condition-maintenance* and *condition-based maintenance*, whereby the terms *fault* and *failure* should be regarded separately from one another. According to [34, p. 2], a failure is the "*termination of the ability of an item to perform a required function*", whereas a fault is considered to be the degradation process or abnormal behaviour that could possibly cause a failure.

While run-to-failure strategies aim at using an asset until its *wear reserve* [49, p. 8] is completely used up, on-condition strategies take into account the realisation of reactive measures for subsystems with low reliability demands and preventive measures for critical technical equipment [17, p. 14]. This can reduce costs arising from longer machine downtimes and the associated production inactivity. Yet, one advantage of the run-to-failure approach is that it fully uses up the useful lifetime of a system [43, p. 20]. However, as this can cause unexpected system failures, it is often more expensive compared to the other two strategies [40, p. 28], as it accepts a high risk of sudden machine failure. To

circumvent this, on-condition maintenance aims at periodically carrying out maintenance procedures, including repair or replacement of machine parts. Although this may save costs caused by production downtime, it is carried out independently of the actual asset condition and thus leads to costs caused by unnecessary maintenance activities. Accordingly, it is assumed that 85 % of maintenance measures are initiated too early and that there are strong dependencies between the asset degradation process and its operating conditions [43, p. 18]. This results in the need for alternatives proving cost-effective solutions while meeting high reliability standards. One solution to this can be to carry out maintenance actions based on the actual condition of the asset rather than on a pre-set schedule [50, preface].

This is the idea behind CBM, which allows to incorporate information about the actual condition of the asset into the planning of the maintenance programme. By using suitable monitoring concepts, fixed inspection intervals can be replaced by dynamically planned maintenance actions [48, p. 16] that aim for a maximum time span between maintenance-related downtimes [47, p. 4].

This concept can help save maintenance costs by reducing machine downtime while utilising the asset's wear reserve to a large extent. Another category worth mentioning has recently come to the fore, namely prescriptive maintenance [51], [52]. This strategy aims at extending the lifetime of an asset by embedding the information about the predicted fault evolution into the ongoing control process. Although some recent studies indicate promising results using this approach, it is beyond the scope of this work.

In addition to the many differences among existing definitions, there is great inconsistency in the use of the term CBM, as it can refer to a maintenance strategy, a maintenance type, or to an information processing model. Thus, the following section provides an overview of the different maintenance types described in the literature. The meaning of CBM in the context of this dissertation will be clarified in Section 2.2.2.

2.1.3 Types

Unlike maintenance strategies, a maintenance type refers more to the specific activities that are performed. The literature shows that most researchers try to classify these types into three categories. Jardine et al. [53] and Cachada et al. [54] have summarized maintenance activities to three maintenance types; unplanned maintenance or *corrective maintenance* [37], *preventive maintenance*, and CBM. As can be seen in Figure 2.3, the DIN 13306 [37] similarly categorises maintenance types, whereby two main types are initially distinguished: corrective maintenance and preventive maintenance. The difference between the two is that corrective maintenance is initiated upon system failures, while preventive maintenance aims to avoid such by taking appropriate measures in advance. Accordingly, this type can be subdivided into *predetermined maintenance* and CBM.

Corrective maintenance comprises the repair activities conducted after equipment failure in order to bring the system back to its original state. Because these tasks are performed in the event of a critical system failure, which can occur at inappropriate times, they often have a high priority and are likely to interfere with other scheduled activities within the production process. In [37, pp. 34-41], it is likewise stated that corrective maintenance can be further distinguished between *deferred corrective maintenance* and *immediate corrective maintenance*. While the former is scheduled after a fault is detected, the latter requires instant action in the event of a machine failure. Since these failures often occur unexpectedly, costs are incurred not only from the actions required to restore

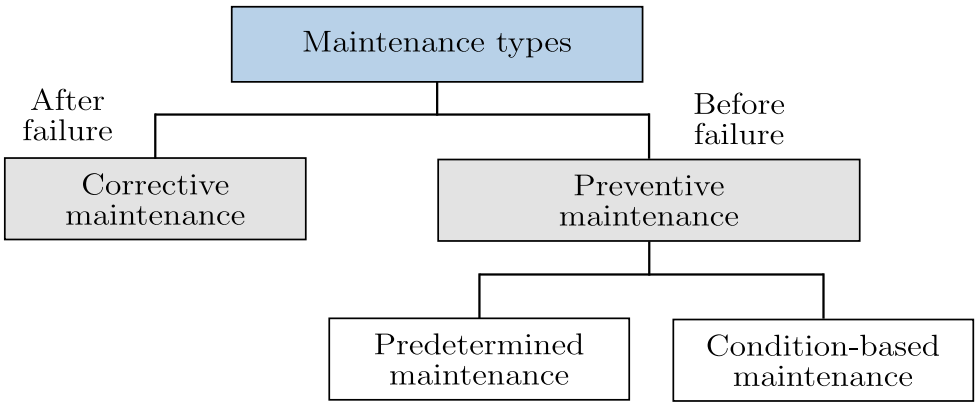

Figure 2.3: The maintenance types according to [37] and [55, p. 819] subdivided according to their preventive measures.

the system to its original operating condition, but also from production line downtime. Dhillen [56, p. 64] describes four downtime components in this context, namely active repair time, administrative time, logistic time, and delay time. Actions to be performed in the context of active repair may comprise the following [57, p. 113], [56, p. 64]:

- Failure recognition
- Fault correction
- Function checkout
- Fault localisation
- Fault diagnosis

Depending on the failed system, the times associated with these actions have different effects on the total system downtime. For example, electrical failures may result in higher troubleshooting times, whereas mechanical failures are likely to cause higher repair times [56, p. 65]. In contrast to corrective maintenance, preventive maintenance involves the planning of maintenance activities in order to reduce the probability of machine failure [37, p. 34]. In fact, the goal is to completely avoid unplanned system downtimes [43, pp. 19-20]. This can be achieved by performing maintenance activities on a regular basis, such as scheduled routine inspections, periodic cleaning, lubrication and overhaul [42, p. 11], which is part of predetermined maintenance. The determination of the maintenance intervals depends on the respective maintenance strategy and can, for example, be performed after a certain operating time or after exceeding a predefined number of load cycles.

When maintenance tasks are scheduled based on the provided condition information of an asset, they are assigned to the maintenance type CBM. Its core concept consists of the accurate prediction of faults and failures as well as the determination of their root cause [58, p. 5]. Although its simplest form only requires human expertise to asses the current degradation state [40, p. 31], modern approaches exploit machine data representing the actual physical state of an asset. This process is commonly referred to as *condition monitoring* (CM) [59], [34], [37, p. 41], which will be described in more detail in Section 2.2.2. By continuously analysing the physical quantities such as, for instance, pressure, temperature or electrical current, it allows to automatically identify changes in the operating conditions [60, p. 44]. As such anomalies can indicate the presence of faults, real-time diagnostic approaches can be utilised in order to schedule maintenance actions on demand. Some techniques even support the identification of future failure risks by predicting the degradation process, thus helping to determine the most favourable time to perform maintenance from an economic standpoint. Although this research area has

attracted many researchers and a large number of scientific contributions exist in the literature, the associated terminologies are rather inconsistent. There are particularly large discrepancies in the use of the terms PM and CBM, which are sometimes used interchangeably. Therefore, in Section 2.2.1, the various terms and definitions in this context will be described and distinguished from each other. In addition, the individual components that drive these concepts will be presented and the scope of this work is explained in more detail.

2.1.4 Measures

Several measures are known to fulfil the maintenance objectives, which can be classified according to their respective tasks. In [49, p. 12] they are divided into: *service*, *repair*, *inspection* and *improvement*. The term service includes all actions that are carried out to delay the degradation process of an asset and to maintain its operational readiness. For example, this may comprise lubrication, conservation or replacement of machine parts or components. If service actions are scheduled on time, they can extend the useful life of the technical equipment and may lower maintenance costs [62, pp. 42-44]. Inspection, on the other hand, refers to asset assessment with regard to its operational condition [49, p. 5]. The aim is to avoid interruptions of the production process by detecting worn equipment before it causes a system breakdown [62, p. 48]. If a technical component has failed or can no longer be operated, repair measures must be carried out to restore the intended functionality of a failed asset [49, p. 4]. Finally, all measures taken to improve the system reliability as well as its maintainability can be ascribed to improvement. It should be noted that such measures are not intended to change the original function of the asset and may involve, for example, system redesign or the identification and elimination of critical vulnerabilities [49, p. 6].

As shown in Figure 2.4, the maintenance type often determines which measures are prioritised to achieve the respective objectives. For example, repair is the key point in corrective maintenance while service and inspection are neglected. In contrast, the other

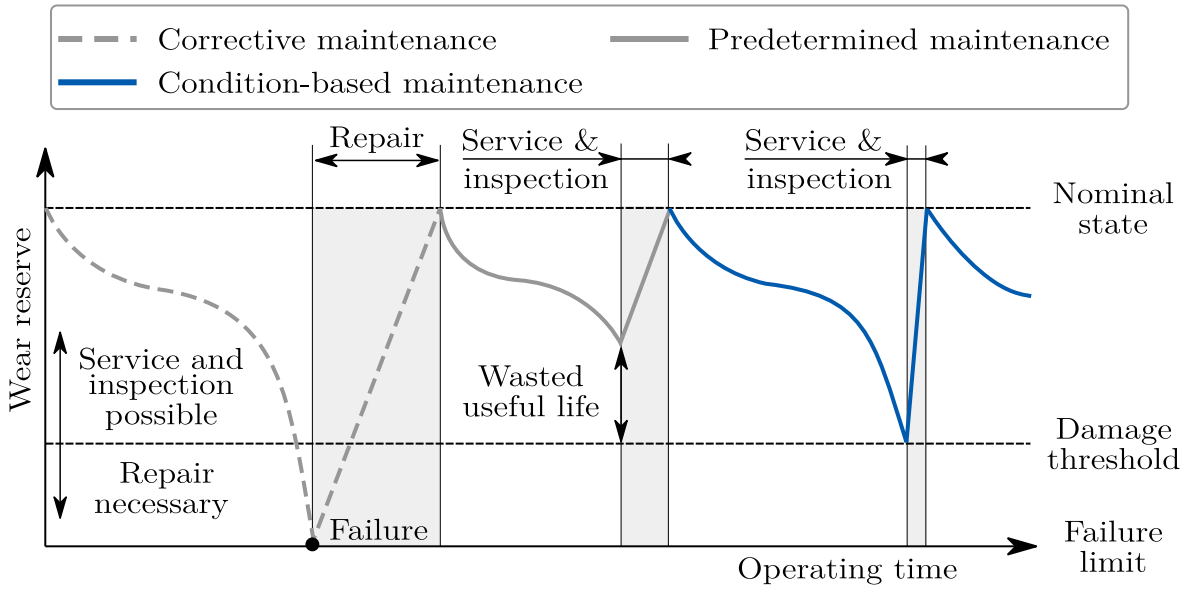

Figure 2.4: Maintenance measures related to different maintenance types during the asset degradation process based on [61, p. 21] and [62, p. 300].

two types primarily focus on service and inspection with the aim of avoiding system failures and, consequently, performing repair actions under high time pressure. CBM, in particular, focuses on inspection [43, p. 18], as it determines the actual condition of the asset, whereupon maintenance actions can be planned. When real-time data are automatically analysed for diagnostic purposes, the time required for manual inspections can be reduced. The reason for this is that the system status can be tracked continuously [40, p. 31] and faults can be located automatically. This makes it particularly attractive from an economical point of view, as human and material resources can be planned more efficiently [39, p. 274].

2.2 Terms and Definitions

Although PM approaches have been a research subject for decades, the related terms and definitions vary widely in the literature. The following sections therefore outline the terminology used in this work and distinguish the content from other fields of activity. First, the term PM is clarified and its components are introduced. The remainder of this section then explains the difference between diagnosis and prognosis and describes CBM in more detail.

2.2.1 Predictive Maintenance

Predictive maintenance covers a wide range of meanings and aggregates multiple sub-disciplines. Surprising as it may seem, it is used inconsistently and repeatedly surfaces within the literature in connection with varying concepts. There are particularly large discrepancies in the distinction between PM and CBM. This is mainly because the former is viewed as a maintenance philosophy that utilises the actual operating condition of a technical system to optimise its overall productivity [47, p. 4], on the one hand, and as a maintenance type, as described in Section 2.1.3, on the other. The main components of a PM concept can generally be divided into two consecutive steps known as *fault diagnosis* and *prognosis* [63, p. 315]. While fault diagnosis aims to provide detailed information about the current system state, such as the type or magnitude of a identified fault, prognosis is concerned with estimating the time to failure [34, p. 12]. It goes without saying that there is a dependency between the two technologies. In fact, the accurate prediction of the remaining asset lifetime is only possible through the use of appropriate fault assessment methods. As outlined in [64, pp. 1-10], this may also be referred to as *health assessment* and its outcome shall indicate the current health state of the physical asset as well as the nature of the monitored fault, including its type or location. This information is then incorporated into the prognosis step to predict the *remaining useful life* (RUL), also known as remaining service life, i.e. the time left before a failure can be recognised [53]. The core of this concept is to estimate the point in time when a component is most likely to fail with the aim of replacing it before it causes unplanned production downtimes.

A more detailed distinction between fault diagnosis and prognosis is carried out in a series of scientific contributions. Accordingly, many authors distinguish between the terms CBM and *prognostic health management* (PHM) to designate these two main steps of the PM information processing chain, such as in [58] or [65]. Similar to the definition given in the beginning of this section, CBM is considered in this context as the estimation of the current fault condition. PHM, on the other hand, is known to represent all fault prognostic and forecasting procedures needed to predict future behaviour including the

RUL [58, p. 13]. Furthermore, it deals with the scheduling of required maintenance tasks and, thus, supports the decision making for the maintenance management. Even though both terms are closely related, they should not be equated. The reason for this is that PHM depends solely on actual machine data, with the aim of appropriately planning the required maintenance activities [58, p. 13], while CBM can also be performed based on human expertise [66, p. 91] and includes, among other things, the actual maintenance activities. To avoid any confusion in the following, this dissertation uses the terms CBM and PHM for their respective tasks, while PM represents the holistic maintenance philosophy.

2.2.2 Condition-Based Maintenance

So far, the term CBM has been introduced several times in varying contexts, more precisely to denote a maintenance strategy, but also as a maintenance type, as has been discussed in Section 2.1. Yet, it is more commonly used to express the information-processing approach to automatically assess the current state of technical equipment, which will be referred to hereinafter. Vachtsevanos [58, p. 13] states that CBM can be seen as the use of runtime data to identify a fault condition. To achieve this, several successive steps are to be taken in the processing of machine data in order to automatically extract fault-related information. The aim is to generate relevant indications that serve as a decision support for determining the optimal time to perform maintenance activities.

An immediate consequence of this is that software for evaluating machine information must focus on data acquisition and processing, which is referred to as *condition monitoring* (CM) [34, p. 1]. In general, this is understood as the measurement of characteristics and parameters [37, p. 41] that allow the actual physical condition of the machine to be assessed. Determining the type of signals to monitor depends on the application and can be viewed from two perspectives: process-related measurements, such as temperature, contamination or tribology [67, introduction], and signal types characterised by high-frequency components, such as vibration or ultrasonic measurements [58, p. 96]. Jardine et al. [53] provide an overview of how machinery data used for diagnostic and prognostic purposes can be classified, which is shown in Table 2.1.

However, extracting useful information from raw measurement data to infer a fault condition requires further steps in terms of data processing and analysis. In an effort to structure these subtasks of a CBM system, many authors have proposed different ap-

Table 2.1: Categories of machinery data according to [53] with examples.

Category	Description	Examples
Value type	Single value logged at a specific point in time representing, for example, one physical quantity	Temperature, pressure, flow rates, humidity
Waveform type	Normally a time series acquired at high frequency data sampling intervals	Vibration, acoustic, ultrasonic
Multidimensional type	Data presented in more than one dimension, which are most commonly images	Thermography images, X-ray images

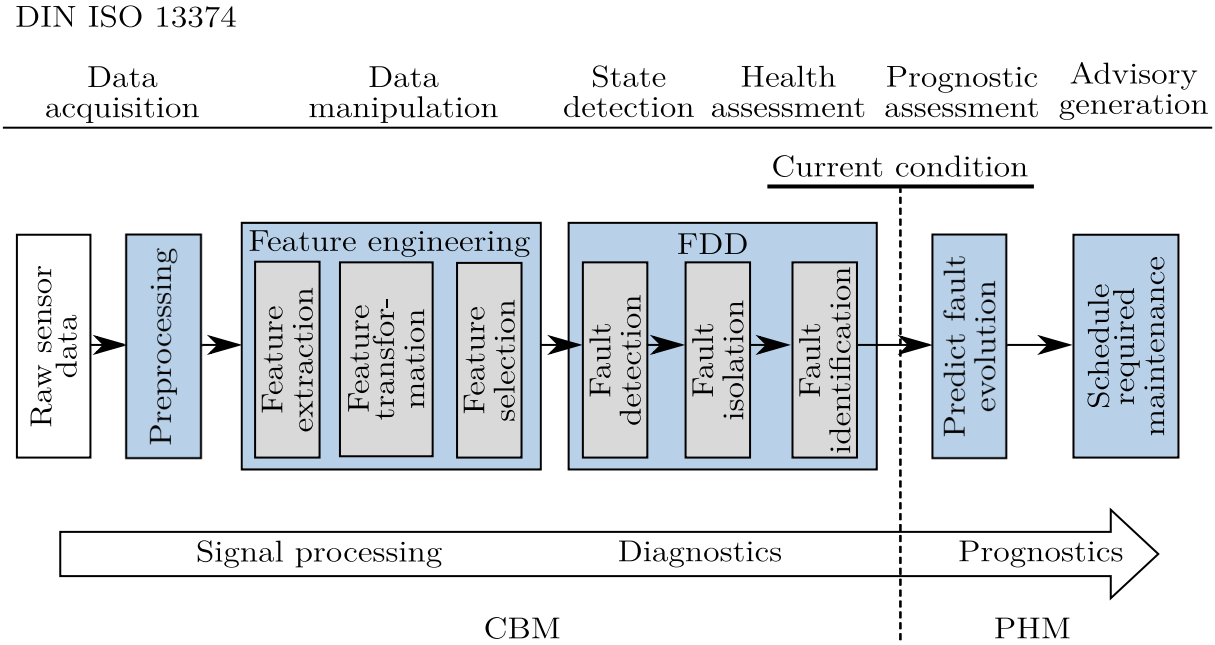

Figure 2.5: Data processing layers within CBM based on [64], [58], [68].

proaches, of which the open system architecture for condition-based maintenance (OSA-CBM) [69] is one of the most widely used. The model developed by the Machinery Information Management Open Systems Alliance aims at providing a detailed communication framework for a wide range of applications by defining seven layers of data exchange conventions that act interoperably and can be exchanged individually [70, p. 52]. Another, but related, view is given within the standard DIN ISO 13374 [64], where the provided framework covers 6 data processing layers starting from the sensory acquisition of machinery data to the support in the planning of upcoming maintenance tasks.

Although these layers are mainly derived from the OSA-CBM model, both guidelines differ slightly in their definitions as well as their purpose. While the DIN ISO 13374 defines a stepwise approach to develop CM based diagnostic systems, the OSA-CBM additionally provides the tools for their implementation [71]. Another framework covering the phases of a CBM system is presented by Vachtsevanos [58, pp. 13-16], who, starting from the collection of relevant data, describes five steps in the information processing chain in accordance with the above layers. In particular, the author refers to the steps *pre-processing*, *feature extraction*, *fault classification*, *prediction of fault evolution* and the *scheduling of required maintenance*, whereas the latter two are considered part of of PHM as outlined in the following section.

In [17, pp. 15-18], both the layers defined within [64] and [58, pp. 13-16] are further particularised to a detailed model representing the entire information process chain of a CBM system. As shown in Figure 2.5, the model covers all steps to be taken to automatically derive the system's health condition from machinery data represented by its sensor readings. It should be noted that this framework serves as the basis for the model presented in this dissertation and is therefore described in more detail below. In addition, further sub-processes of the information processing chain are introduced and the various terms that arise in this context are clarified.

Preprocessing

The aim of signal preprocessing is to extract useful information from the raw measurement by use of both hardware and software components [58, p. 97]. This must be carried out because sensor acquired data is subject to interference, such as measurement noise, and can thus lead to inaccurate diagnostic information [72]. Thus, the most important task of this step is to improve the signal-to-noise ratio (SNR) [55, p. 903], which includes several tools, such as data validation, data compression, amplification or *filtering* [58, pp. 97-103]. Accordingly, data validation is used to verify the correctness of the data obtained and, for example, to detect inconsistencies in the measurement or errors during the data transmission. Amplification, on the other hand, deals with physical quantities whose values can only be monitored at weak signal. The aim is thus to increase the signal's amplitude so that the information it contains can be further facilitated throughout the information processing chain. When developing CBM models for various applications, one may encounter the problem that the amount of data is difficult to handle due to the number of sensors installed or the required data sampling rate, the latter being especially critical when dealing with vibration data, where large amounts of data must often be processed [73, p. 239]. In such scenarios, data compression and decomposition techniques can be applied to reduce the amount of data obtained from the system. Another important tool for improving SNR is filtering, as it aims at de-noising the signal to improve the subsequent data analysis. A typical example constitutes a low-pass filter, since it allows lower band frequencies to pass through the evaluation unit, while higher frequencies, which often represent measurement noise, are discarded [58, p. 98], [17, p. 21].

Another data filtering concept that is especially important in context of this dissertation is steady-state detection, which is used to filter out data stemming from transient system states such as, for example, during system start-up or shut-down periods [74]. As such states indicate high operating dynamics within the monitored system, they may not adequately represent its actual fault condition leading to lower diagnostic accuracy in the following steps. Although few researchers consider the use of transient data in the system health evaluation phase [75], [76], most contributions focus on steady-state data in the development of chiller CBM systems [6]. As will be further described in Section 3.2, exploiting these states for chiller fault diagnostics and prognostics through data analysis techniques is appropriate in many cases, as such systems operate near steady-state conditions for most of the time [6]. Therefore, transient data should be discarded for the data-based health assessment, as this avoids the misinterpretation of the real operating state. Preprocessing may also comprise the estimation of variables that cannot directly be measured. However, one may be able to measure related quantities and then draw conclusions about the quantity of interest [58, p. 97].

Feature Extraction

Feature extraction, or sometimes called feature generation [68, pp. 1-2], is applied to infer more relevant information from the available data [55, p. 906] with the aim of accurately assessing a fault condition if present. Hereby, the term feature is known to be an attribute or variable that describes certain aspects of any data object [68, pp. 1-2]. The main goal of this phase is to improve the accuracy of the prediction step [77] by deriving additional information from the dataset. Especially with regard to CBM, these features should be highly sensitive to faults [6] in order to ensure a reliable assessment of the system

Table 2.2: Overview of the different feature types with examples.

Feature Type	Examples
Engineering knowledge features	Coefficient of performance, heat flow, pressure difference, enthalpy, polytropic efficiency
Time Series features	Lagged features, derivatives
Statistical features	Average, standard deviation, mode, median, range

state. As outlined in [55, p. 906], three domains can be defined in which features may be extracted, namely time domain, frequency domain and time-frequency domain. While the former focusses on processing the data sampled at certain points in time, such as temperature measurements [58, p. 97], the latter two are more about analysing vibration data that require the transformation of data in a new domain by applying for instance fast Fourier transformation [55, p. 906]. As will be described in Section 4.1, the work at hand exploits process data in the time-domain, which is why the data transformation into the frequency domain or the time-frequency is beyond the scope of this dissertation.

In [78, pp. 18-20], the extraction of features within the time domain and with respect to fault assessment is divided into the three categories: engineering knowledge features, time series features and statistical features. Accordingly, deriving features based on engineering knowledge involve specific domain expertise and therefore require an overall understanding of the field of application. This takes on even more importance in terms of chiller CBM, as many thermodynamic quantities, such as enthalpy or entropy, cannot be measured by use of standard sensors but can be estimated through, for example, arithmetic operations or refrigerant property tables [6]. As listed in Table 2.2, time series features take into account the previous observations from which additional information can be obtained. This can be the time derivative of a certain signal or simply the difference of a measurement between two or more time intervals, which eventually provide additional information about the state of a technical asset. Lastly, statistical features can be derived by analysing the given dataset, such as determining the feature average in a predefined time period.

Feature Transformation

Feature transformation is often associated with feature extraction and sometimes even used as a synonym for it, as both approaches can refer to the construction of additional features [79, p. 189] and are therefore often performed in a single step throughout the information processing chain. Nonetheless, they should not be equated. This is mainly due to the fact that feature extraction aims at generating new features from the given data pattern, which may also include domain-specific computations, while feature transformation is more about defining some sort of mathematical mapping of the dataset that benefits the prediction process. In other words, features generated throughout the feature extraction phase are not the result of feature transformations [68, p. 3]. There are several ways to find a mathematical mapping that leads to a new data representation of which *principal component analysis* (PCA) or linear discriminant analysis (LDA) belong to best known ones [80, p. 153]. Section 3.1.3, provides an overview about the mapping methods used in dissertation and explains their working principles in more detail.

Another important feature transformation procedure is data scaling, which is also often attributed to the feature transformation step [80, p. 151]. This is especially important for

data-driven approaches in order to prevent particular variables from dominating the fault assessment approach [81, p. 14]. As will be pointed out in Section 3.4, multiple scaling methods are described in the literature, of which normalisation and standardisation are most commonly used [82, p. 19]. Even though scaling is usually understood to be part of the preprocessing layer [17, p. 21], in this work it is performed within the feature transformation layer for simplicity reasons. This is mainly due to the fact that some features rely on substance-specific parameters that depend on one or more original values of the sensory recorded physical quantities for their calculation.

Feature Selection

After generating further relevant information from the original variables, feature selection methods should be applied to find a smaller feature subset for the final prediction step [78, p. 20]. In fact, the primary goal of this step can be expressed as minimising feature redundancy on the one hand, and maximising feature relevance on the other [83]. This can have a positive effect on the processing of larger amounts of data, as it reduces the amount of exploited features and, thus, leads to lower computing times [78, p. 20]. Even more, it may enable the use of certain algorithms [68, p. 3] in the subsequent phases, which would otherwise be technically impossible or only possible to a limited extent.

Especially with regard to CM, this can also mitigate the cost of implementing a CBM system by reducing the number of necessary sensors [74]. Another convenient aspect on feature selection is that it could also improve the predictive quality of the applied algorithms [68, p. 3], as it allows to cut down the available data to the most essential components. This addresses a problem that often arises in the context of ML, namely the *curse of dimensionality* [68, p. 118], [17, p. 25], which describes a phenomenon in which the generalisation ability of the predictor is reduced by high data dimensionality in combination with small amounts of available observations [84]. The aim of this step is therefore to discard features with low information density, e.g. by discarding features with zero variance [78, p. 20] or features that are highly correlated [85]. Several methods exist to accomplish this: on the one hand, important features may be selected through expert and domain knowledge, on the other hand, this can be done through the use of appropriate metrics placing data variance and entropy [17, p. 28] at the center of consideration. Another common approach is to use sequential search methods [86, p. 21], which aim at evaluating the information density of the available feature representation by monitoring the change in the algorithm's predictive performance as features are added to or removed from the dataset.

Particularly with view to chiller CBM, many researchers stated that by deploying certain characteristic quantities for data-driven health assessment purposes, the applied algorithms can be more effective [6]. Thus, various approaches have been pursued in this area comprising the selection of features through search methods based on, for example, evolutionary algorithms [85], domain knowledge [6] or by identifying fault sensitive subspaces after performing PCA on the available data [87]. Further work on this is presented and summarised in Section 2.5.

Fault Detection

To automatically retrieve the actual system condition, fault classification must take place, to which in the following will be referred to as *fault detection and diagnosis* (FDD).

It should be noted that the term FDD is often used as a synonym for all phases of the CBM model to estimate the current health state of the system, as in [74] or [85]. In this work, however, this rather comprises the three steps to infer a fault condition, as [58, p. 176], [53]: (1) fault detection, (2) fault isolation and (3) fault identification.

In the first layer, the deviations from the normal operational state of a system must be identified. Thus, the aim of fault detection is to decide whether a fault is present or not [88, p. 14] and to draw attention to any abnormal operating conditions [59]. In most cases, it is a binary statement that represents the presence of anomalies and gives no indication of causes or specific faulty components. To this end, several approaches are known that enable the detection of faults at an early stage of development. A widely applied approach is to monitor the difference between a reference model and the actual process. By selecting a specific subset of features that indicate the presence of the faults to be monitored, one can define some rules that support the automatic differentiation between normal and abnormal operating condition [53]. Other considerations relate to the use of statistical, or *machine learning* (ML), models that rely on historical machine data for their development [58, p. 9]. Further explanations regarding the classification and working principles of fault detection approaches are provided in Section 2.3.

Fault Isolation

In the second step, fault isolation is performed to identify the faulty component or subsystem [53]. While some authors consider this step as a procedure for narrowing down the number of components affected by a particular fault, others also interpret the determination of the respective fault type as its constituent part [17, p. 26], which is also considered in this dissertation. Therefore, the main task of this layer is to perform an accurate classification of the corresponding fault after abnormal behaviour could be detected, for which a wide range of approaches exist. For example, decision tables that relate predetermined characteristic features to a certain fault [6], or ML models that capture specific fault patterns can be exploited for fault classification tasks [74]. Yet, this process relies on the early detection of abnormal system behaviour and is therefore often directly combined with the former step.

A major challenge in the development of such systems is that a wide variety of operating conditions can prevail within the system to be monitored, which can lead to wrong system diagnostics if the underlying model design does not capture these dynamics or external influences. The goal in developing such systems is to provide a scheme that is both robust to uncertainties and sensitive to faults [89, p. 5]. Moreover, it should accurately identify impending or incipient faults by simultaneously providing a low false-alarm rate [58, p. 176].

Fault Identification

In the final FDD step, fault identification is conducted in order to determine the extent and the nature of the identified fault [58, p. 177]. The former is also known as *severity estimation* and is particularly important for the subsequent PHM system, as it forms the basis for the prediction of the future degradation process [17, p. 26]. According to [64], the outcome of the health assessment phase, and therefore of fault identification layer, should be the actual system condition described by a health index ranging from 0 to 10, whereby 0 represents a system failure and 10 a fault-free state that

might be present after commissioning or directly after maintenance has been performed. Still, estimating how far a fault has progressed can be a difficult task, as it is subject to a number of uncertainties. In some reports, however, approaches can be found which apply, for instance, statistical property and residual signals [90] or distance metrics after applying a classification algorithm on the provided data [91] for severity level estimation. This is particularly important for the subsequent PHM system, as it forms the basis for the prediction of the future degradation process [17, p. 26].

2.2.3 Prognostic Health Management

After determining the current system state through the information processing within the layers described in the previous section, a PHM system is used to schedule the best possible maintenance time. As illustrated in Figure 2.5, this is considered the fault prognosis scheme and consists of two consecutive steps: *predict fault evolution* and *schedule required maintenance* [58, p. 15].

Predict Fault Evolution

The first layer within the PHM process deals with the prediction of the future system degradation process by focusing on the estimation of the RUL [58, p. 13]. To this end, it uses the information provided by the FDD block, which includes both the type of fault and its severity, using the health index as an indicator. In the event that a fault is present in the system under consideration and its severity can be reliably estimated, the task of this step is to determine the time at which the identified fault causes a failure, i.e the RUL [64]. Ideally, the output of this layer should provide a reliable estimate of when the wear reserve of the monitored device will be completely depleted so that it can be replaced at the most economically advantageous time [50, preface]. However, since this might not be practically feasible due to the many uncertainties along the fault assessment process, this layer should also provide uncertainty bounds indicating the confidence in its prediction [58, pp. 285-314].

Schedule Required Maintenance

The final layer supports the decision making based on the analysed system state by providing actionable information in terms of performing maintenance actions or changing the operational condition of the considered system [64, p. 3]. Thereby, the scheduling of the required maintenance tasks depends on the overall maintenance strategy of an organisation and may thus, vary according to its needs.

Although PHM is indeed important for planning cost-effective maintenance, the present work focuses primarily on assessing a chiller's current fault condition while following all phases of the CBM scheme. The reasons for this are twofold: firstly, accurate fault diagnosis is a prerequisite for the integration of a PHM system and therefore special attention must be paid to the associated phases. Secondly, as will be outlined in more detail in Section 2.4.3, there are particularly high challenges for deriving chiller CBM models due to vastly varying system designs and operating conditions, which complicates their development. Therefore, a core aspect of this work is a novel approach towards a reliable model that overcomes these challenges while offering significant cost benefits for use in industrial refrigeration systems.

2.3 Approaches

The development of CBM systems has been the subject of research for decades, as a result of which many approaches exist today in various engineering disciplines. As a consequence, the models underlying these systems are subject to different theoretical perspectives and use a variety of techniques to fulfil the respective domain-specific task. To this end, this section provides an abstract overview on the fault evaluation task and introduces the concepts exiting in the literature. The remainder of this section then outlines the categories into which these approaches can be divided.

2.3.1 Overview

The aim of a CBM system is the accurate fault assessment of the monitored asset by continuously observing its system parameters. Following the data processing sequence presented in Section 2.2.2, the respective model analyses the current system state represented by both the set-point variables and the sensor readings. In [88, p. 13] this is depicted as follows (see Figure 2.6): Let u be the set of input variables and $\dot{x}$ the measured output of the monitored system affected by disturbances d and some fault fti (later referred to as a fault type index). At a specific point in time, the model evaluates the system health state by observing u and $\dot{x}$ and estimates the fault type fti as well as its severity level represented by a health index hi. As will become clearer at a later stage of this work, u must not necessarily be an input to the CBM model.

Even though this well describes the primary goal of a CBM model, it does not illustrate the methods used to extract fault-related information from the system. As it has been mentioned in Section 2.2.2, the acquired data are processed through multiple layers to assess the current fault condition, with each layer serving a different task. Throughout the years, general research has predominately been focussed on FDD, as the most critical part is to detect and isolate faults at an early stage of development [58, p. 3]. This has led to a large number of scientific contributions in this field considering chillers as well as other technical equipment, which can generally be classified into model-based and data-based, or sometimes also called model-free, approaches [92].

According to Zhang et al. [92], each of these schemes can further be classified into quantitative and qualitative approaches leading to three more detailed categories. As

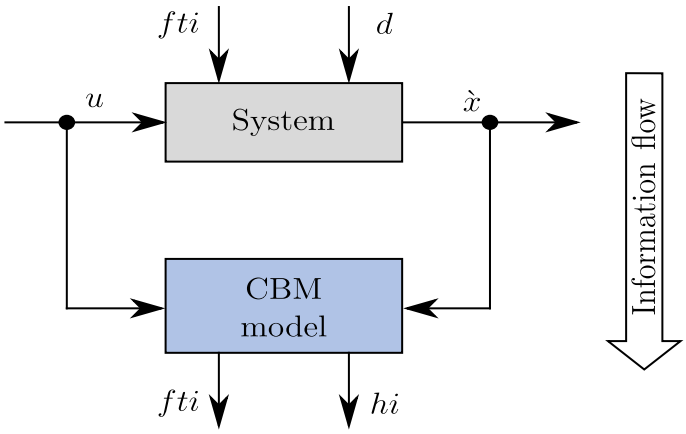

Figure 2.6: General approach towards CBM showing a system affected by disturbances and a fault as adapted from [88, p. 13].

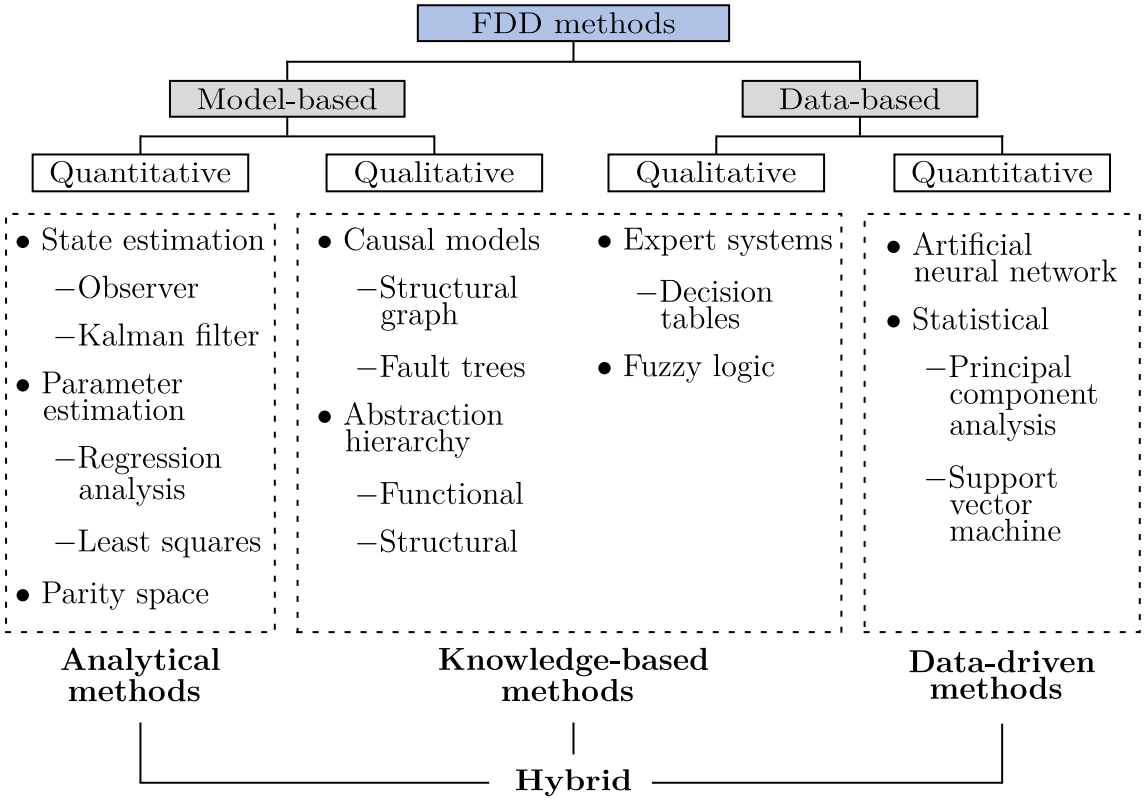

Figure 2.7: Overview FDD approaches divided into model-based and data-based following the suggestions of [92] and [93].

shown in Figure 2.7, quantitative model-based approaches are also known as *analytical methods* and rely on an accurate dynamical model [58, p. 178] to perform real-time FDD [92]. *Knowledge-based methods*, in turn, can be assigned to both model-based and data-based approaches and are qualitative in nature in both cases. These methods usually rely on domain-specific knowledge in the form of engineering experience or historical failure cases and often represent an alternative to analytical methods, especially when a detailed process model is not available [93]. The third type of method includes quantitative data-based approaches, which are referred to as *data-driven methods* [94]. Approaches falling in this category often deploy methods from the field of ML [95] and can thus benefit from the availability of large amounts of machine data [6] to capture the fault intrinsic relationship. Nonetheless, some proposed FDD approaches do not fall into just one of these categories but rather combine them, which are known as *hybrid methods* [96].

2.3.2 Analytical

The underlying concept of analytical FDD methods is that these approaches aim at replacing hardware redundancy with a software-based process model that adequately represents dynamic process behaviour [97, p. 6]. During operation, this model is used to generate residuals between the artificial output and the actual system output Y that may indicate the presence of a fault [58, p. 179]. Throughout many scientific contributions, the model simulating the real system condition based on some input information U is often designated as the *reference model* [24], [97]–[101]. As it aims to predict normal (fault-free) system behaviour, it allows the monitoring of consistencies with the observed process.

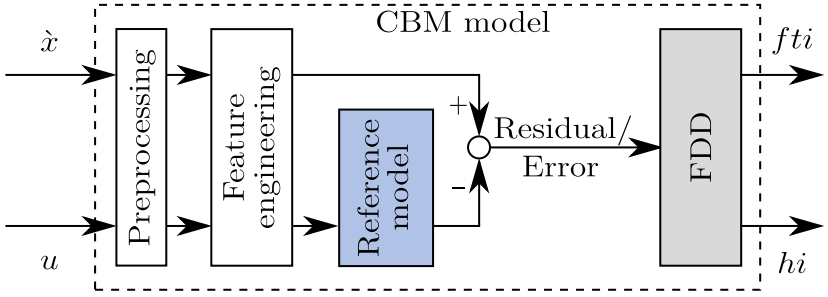

Figure 2.8: Analytical approach towards FDD as derived from [94] and [17, p. 27].

If a fault is present, deviations between the reference model and the process are to be expected on a larger scale, allowing faults to be detected [102]. Several methods are known for deriving such a model, including state estimation, parameter estimation, parity space evaluation, or a combination of these [92]. As shown in Figure 2.8, the residuals of some predefined features are then evaluated within the FDD block in order to deduce the current fault condition, which can i.a. be archived through the use of some predefined residual thresholds.

In general, analytical methods require the construction of a mathematical model, usually created through the use of first principles, and therefore often incorporate a physical understanding of the system [93], [103, pp. 6-10]. However, defining an accurate model being capable of capturing all dynamic relationships can be a very challenging, if not infeasible, task, especially when the dependencies of the monitored process are highly non-linear [89]. Thus, these types of methods may not be an adequate choice for systems with high complexity [94] due to the increased effort in the development phase. If, on the other hand, analytical methods are applicable, they are often superior to the methods assigned to the other two categories and should therefore be preferred if the physical dependencies can be suitably expressed in mathematical form [103, pp. 6-10]. Despite the reference models aiming to simulate the physical system behaviour, some authors also tried to apply both linear [24] or non-linear [99] regression models such as, for example, support vector regression [104]. This leads to another categorisation approach of reference models that, according to [105], can be divided into *black-box* (regression) and *white-box* (first principles) models.

2.3.3 Knowledge-Based

It is understood that analytical methods have their limitations, as they rely on an accurate dynamic and sometimes highly non-linear model. One way to cope with this problem is to apply knowledge-based methods that define a set of rules [93] to infer fault information from system parameters. However, these methods are mostly used for fault isolation rather then for fault detection as, for example, in [6], [99] or [106]. Thus, many authors apply either analytical or data-driven methods for fault detection tasks, and isolate their root cause through some predefined rules, which can generally be assigned to both qualitative model-based and data-driven approaches [93], [103, pp. 6-10], as shown in Figure 2.7. While qualitative model-based methods comprise i.a. abstraction hierarchy [107] or causal methods, such as fault trees [93], qualitative data-based methods are based on the evaluation of real-time machinery data [108]. Today, most of the knowledge-based FDD methods fall within the latter category and are usually based on rule-based expert

systems [109], such as decision tables [6] or fuzzy expert systems [110]. In these cases, one attempts to implement the operator's or engineer's domain-specific knowledge into decision-making software, including information about input and output variables, abnormality conditions, fault symptoms or operational constraints [108]. These methods can be especially applied if an analytical model is not available or difficult to obtain and if the number of input and output variables is comparatively small [103, pp. 6-10]. Despite that, with the rapid growth in computational power and the increasing amount of FDD related software packages being available, these methods become more and more suitable for complex problems [109]. Although they can greatly benefit from such human level knowledge and often indicate their robustness for fault isolation tasks [96], they may suffer from the lack of detailed fault information. Because the deduced rules are based on historic failure cases as well as experience, it may be difficult to infer new fault types that are beyond the scope of the available expert knowledge [109].

2.3.4 Data-Driven

Due to the rapid development in the field of ML, data-driven models have particularly gained the attention of researchers in recent years. In essence, these methods exploit machinery life-cycle data [103, pp. 6-10] by capturing the intrinsic and fault related relationships underlying the data. This is especially beneficial for FDD scenarios in which the monitored technical system has non-linear properties [92] or when one has to deal with high dimensional datasets [95]. As these methods neither rely on a reference model such as, for example, based on first principles, nor require expert knowledge, they are best suited for large and complex system architectures and can therefore save time and costs in their development [103, pp. 6-10].

As shown in Figure 2.9, the core of a data-driven FDD method forms the *trained fault model* that is developed by use of historic system data. This model may consist of one or multiple classifiers that, during the development phase, aim to find well describing decision boundaries between various fault patterns [111], e.g. for each fault class fti respectively, which is hereinafter referred to as the *training phase*. In this regard, many approaches

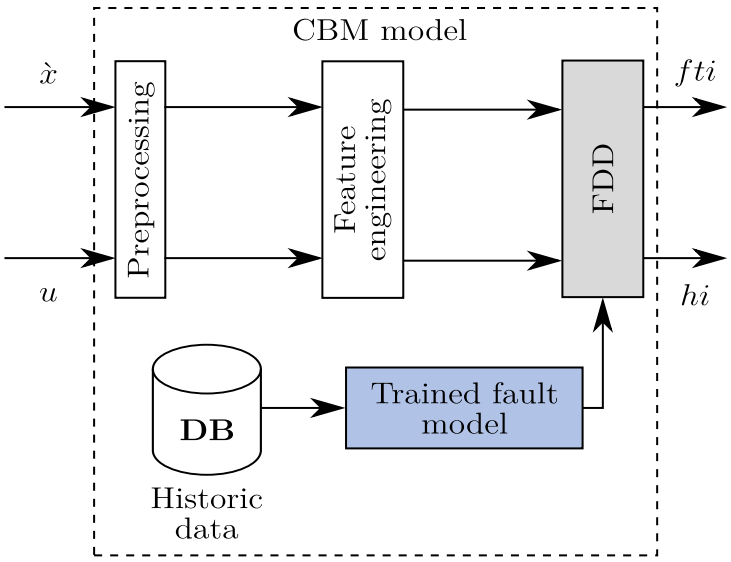

Figure 2.9: Data-driven approach towards FDD [17, p. 28], [58, p. 179].

exist in the literature that greatly vary throughout the various domains and applications and can generally be divided into *supervised*, *unsupervised* and *semi-supervised* ML [112, pp. 10-12]. While supervised ML aims to exploit the respective class labels by inducing prior knowledge about the target output [53] during the training phase [58, p. 108], unsupervised machine learning allows the use of data patterns that are not directly assigned to a specific fault [94].

Following these definition, it appears that unsupervised learning may be advantageous compared to supervised machine learning in terms of FDD. The reason for this is that these approaches can omit time-consuming data labelling, e.g. assigning each observation in the dataset to a specific class representing a system condition, such as fault-free, faulty or the respective fault type [95]. This is, however, somewhat misleading because unsupervised machine learning models are also often limited in their application. For example, they are more likely to find hidden similarities across the data, such as in clustering [95], or to reduce the data dimensionality [91]. As a consequence, they are often less suitable for fault isolation tasks, as automatically assigning an unknown observation to a specific fault type requires associated labels to be exploited in the model training phase [91]. On the other hand, unsupervised machine learning may be applied in the feature engineering block or in the fault detection phase of the CBM scheme. The latter applies to the case where normal samples can be assumed to be more frequent than faulty ones, which allows their use for anomaly detection tasks [112, pp. 11].

In some cases, however, one encounters the problem that only a limited number of the available observations are labelled, while the rest of the dataset remains unlabelled. In this way, semi-supervised machine learning can be applied to exploit the unlabelled data patterns within the minority labelled training dataset to enrich the overall information content [113]. This can also be beneficial since supervised machine learning assumes all data to be correctly classified before training the respective classifier, which cannot always be guaranteed due to uncertainties during the labelling process. Therefore, semi-supervised approaches can be used to improve the overall classification accuracy by embedding unlabelled data in the training process [96]. Such models are often superior to supervised machine learning models, as it is usually difficult to obtain training data covering all facets of anomalous behaviour that can possibly occur [112, p. 11]. Another but strongly related concept is *one-class classification* (OCC) [35], where only one class is accurately labelled. One well known algorithm used for fault detection tasks is, for example, the one-class support vector machine (OC-SVM) [114], as applied in [91] or [85], where only a small amount of normal labelled samples and a large amount of unlabelled data are available. Note that this algorithm can be used in both a semi-supervised as well as an unsupervised setting. While the former is primarily used for *novelty detection*, in which only some labels of the target class are available, the latter applies to find *outliers* with no explicit labels being used in the training phase [115, pp. 181-188]. As a consequence, different types of algorithms exist, which are subject to varying concepts and mathematical formulations. Table 2.3 provides examples on some algorithms assigned to their respective category in the context of FDD. However, even though most algorithms may fall in one of these three categories, there are also other ML approaches, such as *transfer learning*, which aims to circumvent expensive and time consuming data labelling by transferring knowledge from one *learning task* or *domain* to another [116]. As will be outlined in Section 3.1, this methodology is also applied throughout this dissertation.

Although the use of data-driven FDD models is beneficial in many respects, there are also some shortcomings that should be addressed. As mentioned by many authors [6],

Table 2.3: Examples of algorithms used for data-driven FDD.

ML Type	Algorithm	Source
Supervised Machine Learning	Artificial Neural Network	Han et al. [117]
	Support Vector Machine	Han et al. [74]
	Nearest Neighbour	Jardine et al. [53]
	Random Forest	Shrivastava et al. [118]
Unsupervised Machine Learning	Association Rules Mining	Yairi et al. [119]
	K-Means Clustering	Jung et al. [96]
	One-Class Support Vector Machine	Harrou et al. [120]
Semi-supervised Machine Learning	PCA T^2 Statistic	Beghi et al. [6]
	Generative Adversarial Network	Yan et al. [30]
	One-Class Support Vector Machine	Beghi et al. [91]

[53], [101], [103], [112], the lack of available labelled data often inhibits the successful application of data-driven methods in this area. In addition, some models may not be robust to rapidly changing system operating conditions or unknown fault situations [6], resulting in lower health assessment accuracy. Nevertheless, with increasing amounts of data [14] being available from various domains, data-driven methods have gained rising interest in recent years due to their ability to capture complex relationships in the data provided [109] and because they can be developed regardless of a physical understanding of the respective system [121].

2.3.5 Hybrid

As the three aforementioned methods have specific strengths and flaws, some authors have also tried to combine these in order to archive better classification performance [95]. While hybrid approaches allow to take advantage of both model-based and data-based approaches, their development is often motivated by the lack of fault specific training data, which complicates the exclusive use of data-driven methods [29]. The most frequent case of hybrid methods in the context of this work is fault detection, where analytical and a data-driven models are combined in such a way that the former are used to generate fault-specific residuals, while the latter provide alarm-triggering thresholds, as described in [96]. Other authors, in turn, applied analytical methods, such as filtering techniques [85], to stationarise the time series data from a certain system and to improve the SNR. Then, a classifier is trained using the new data representation to identify the respective fault related decision boundary. Although combining multiple methods for FDD brings new opportunities, there exist also some challenges in their development. One problem is that aggregating and fusing heterogeneous information stemming from different models may be a difficult [122] and time-consuming task. Furthermore, when combining several methods, one can benefit from their advantages, but also inherit their disadvantages. If, for example, an analytical model should be utilised in a hybrid fashion, it remains questionable whether it can be obtained at low development costs [123].

2.4 Application

The following section provides an overview of the system that is the scope of this dissertation and clarifies the type of related faults that can possibly occur. For this purpose, the vapour compression refrigeration cycle and its basic thermodynamic principles are first introduced. Secondly, the associated fault categories are presented with corresponding examples and distinguished from each other. It is also clarified what types of faults are particulary considered throughout this work. The final part then deals with the challenges arising from the development of suitable CBM systems for chillers.

2.4.1 Principles

Among the existing refrigeration technologies, vapour compression refrigeration is one of the most commonly [125, p. 61] used in both commercial as well as industrial applications. These systems exploit refrigerants as working fluid with the aim of removing heat from a chilled system [124, p. 272]. To this end, such systems take advantage of the latent heat released or absorbed during a phase transformation [126, p. 32], such as in boiling or condensation. According to [124, p. 272] and [125, pp. 61-75], the idealised refrigeration cycle is described as follows: As shown in Figure 2.10, the compressor draws gaseous refrigerant from some suction pressure p_{re} with an associated temperature T_{re} and compresses it, which requires work w_t, in the state transition 1-2 leading to a higher pressure p_{rc} and temperature T_{rc}.

In the ideal case, this state transition takes place isentropically, i.e. reversibly and adiabatically, and thus under constant entropy. This, however, is not necessarily true for real world processes, since irreversibilities occur during the state transitions along the refrigeration cycle causing a rise in the entropy. In the following step, heat is transferred from the refrigerant as it passes a heat exchanger, hereinafter referred to as the condenser, while the vapour is liquefied. The heat Q_C to be dissipated from the refrigerant is transferred to another medium, such as water [125, p. 62]. In the state transition 3-4, the working fluid is expanded while the specific enthalpy h_e is maintained. Finally, the refrigerant evaporates within the evaporator while absorbing heat Q_E from the chilled system.

In HVAC&R terminology, the system providing the cooling load (chilled water or air) is called the chiller and primarily consists of the aforementioned subsystems compressor, evaporator, condenser and expansion valve [127, p. 10]. Nonetheless, further parts are needed to ensure the functionality of the chiller including oil and refrigerant separators, water pumps or oil recirculation systems. Although the chiller itself is also a subsystem of a large-scale HVAC&R system [85], which may consists of further heat-exchangers, pipes, air handling units or rooftop units [128], it is considered the most critical part, as it is responsible a significant amount of the overall energy consumption [129]. Thus the scope of this dissertation is placed on chiller systems with special view to industrial applications. Besides, the focus of the work at hand is on the vapour compression technology, while other types, such as absorption chillers, are not specifically addressed. For more details regarding the design and operation of refrigeration systems, the reader is referred to [124], [125], [127].

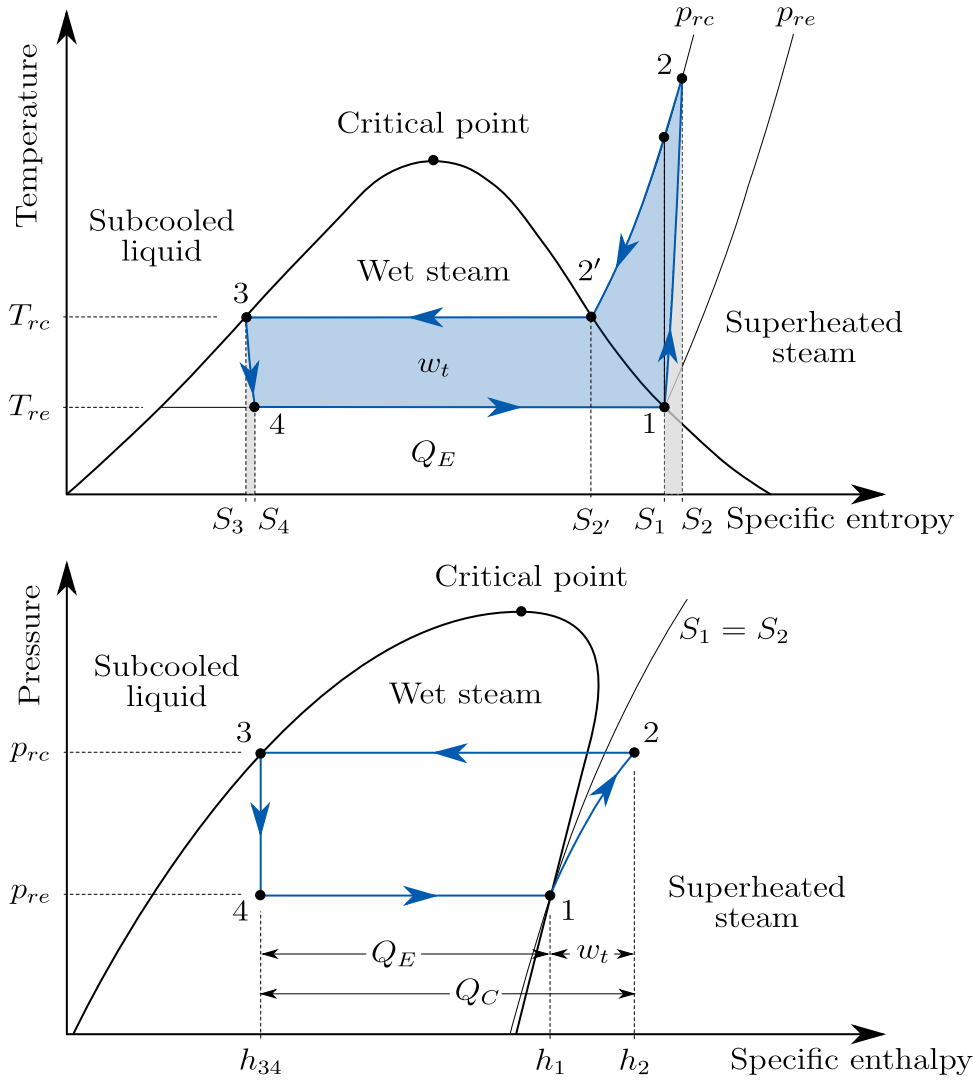

Figure 2.10: Refrigeration cycle as derived from [124, p. 272] showing the scheme of a T-S (temperature-entropy) diagram (upper) and the scheme of a p-h (Mollier) diagram (lower).

2.4.2 Fault Classification and Delimitation

As with all technical equipment, one or more faults may occur that affect the overall functionality of the respective system. These can generally be divided into *hard* and *soft faults*, whereas hard faults usually occur abruptly and may cause the system to stop functioning [130] and can, thus, also be referred to as hard failures (see Section 2.2.1 for a detailed distinction between faults and failures). On the other hand, soft faults usually lead to a progressive degradation of system performance [131], while at the same time, however, continued system operation is possible. In the context of the present work, a hard fault is, for example, a compressor failure, while a soft fault can be caused by dirt accumulation in heat exchangers [130] or refrigerant losses [131]. According to [130], hard faults are typically easy to detect and to isolate because their occurrence is obvious and they cause an immediate stop of system operation. On the contrary, the reliable detection and isolation of soft faults is fraught with difficulty, as the affected chiller system may appear fully functional to the operator [23] while its overall performance is

already degraded. Despite these two categories, many authors also distinguish between *system-level* and *component-level* chiller faults [23], [117], [130], [132], whereas the former fault types are not limited to a specific location [133, p. 11] but rather affect the holistic system. Wang et al. [23] further divide system-level faults into sensor faults, where there may be erroneous sensor readings or measurement bias, and process-related system faults. In contrast, component-level faults affect only a certain component or subsystem including, for example, mechanical or electrical units [133, p. 11] from which other effects on the system may originate [132].

In a study conducted by Comstock et al. [133], the authors identified the most critical faults occurring in chiller systems, of both hard and soft type, by including five chiller manufacturers and a total number of 508 service reports in their study. The investigated faults and failures were ranked according to their frequency of occurrence as well as their cost impact. From the faults investigated during their survey, the authors concluded that not all of them are critical in the development of CBM systems. This is mainly due to the fact that many faults can be detected with little effort and simple equipment. In contrast, soft faults related to system degradation processes are not easy to detect and consequently special attention must be paid to these types. The survey concluded that the following faults are the most critical due to their difficult detection characteristics, some of which are also considered in this dissertation:

- Reduced condenser water flow
- Refrigerant leak
- Excess oil in the system
- Non-condensables in the refrigerant
- Reduced evaporator water flow
- Defective expansion valve
- Condenser fouling

Although some of these faults did not occur frequently during the data collection phase of the study, they accounted for about 42 % of the overall performed services and resulted in 26 % of the total repair costs. It should be noted that in addition to repair-related costs, there may be additional costs due to the reduction in the coefficient of performance ε, resulting in higher overall energy consumption. Consequently, their early detection as well as their root cause identification can be of high value, which increasingly emphasises the use of sophisticated CBM systems. As will be described in more detail in Section 2.5, accurate assessment of such faults can be achieved by analysing the thermodynamic conditions of the respective chiller [133, p. 25] as represented by its sensor readings.

2.4.3 Problem Description

A major challenge in the development of chiller CBM models, or FDD in particular, constitutes the high system complexity as well as the multitude of different system architectures and operation conditions, as this complicates the nature of faults [102]. In particular, chillers utilised for industrial purposes are often custom-built [26, p. 464], while they are developed for specific application requirements. The energy- and cost-efficient design of the system is primarily based on the actual application and determines both its structure and its operating conditions. This results in greatly varying system architectures being applied for various cooling tasks. One examples is the choice of working fluid. While older systems often utilise chlorofluorocarbon or hydrochlorofluorocarbon-based refrigerants, which are known to contribute to global warming or ozone layer destruction when released into the environment [127, p. 9], newer systems are mostly based on environmentally friendly refrigerants, such as ammonia (R-717) [125, p. 65-69]. Other factors

include the selection and design of individual components, such as heat exchangers or the compressors used, as well as the addition of further modules, such as components for storing thermal energy, the introduction of multi-stage refrigeration cycles [125, p. 85], the required cooling load or the ambient conditions.

This suggests a general problem arising from the highly differing systems and applications in the development of chiller CBM models, as they are usually developed for a dedicated system only [29]. On the other hand, deriving a model that generalises well to a wide range of chiller operating characteristics and architectures is not a trivial task, due to the high technological complexity involved [29], [132], [134]. This especially emphasises the motivation of this work for mainly two reasons: firstly, to develop a model with reliable fault assessment capabilities, and secondly, to propose a cost effective solution by avoiding the CBM model development from scratch for each system.

2.5 Related Works

This section provides an overview of the state-of-the-art CBM models and highlights the latest research work carried out in this field. Existing models used for chiller fault assessment are reviewed and their results are compared, whereby their transferability to heterogeneous systems is evaluated. Finally, the presented scientific contributions are summarised and the open research gap is discussed in detail.

2.5.1 Models

CBM, and more specifically FDD, has been subject of research for years and many scientific contributions exist throughout the literature with application across many industrial sectors. Especially chiller CBM has been extensively studied as it has been shown to offer great opportunities for energy and cost savings [6], [87], [135]. To this end, the topic related models are presented and assigned to their respective working principle in the following, which has been introduced in Section 2.3.1. It should be noted that the focus of the following section is on contributions concerning FDD published within the last decade. In general, deciding whether a model is to be assigned to a particular category tends to be fuzzy because authors often mix several FDD approaches in their work. Consequently, the following models are categorised based on their novelty detection principle when both fault detection and fault isolation are addressed in the respective paper.

Analytical

First attempts to integrate chiller FDD strategies have led to the development of white-box reference models aiming to find a mathematical formulation to describe the thermodynamic and fault related dependencies, as introduced in Section 2.3. Some recent work on this is presented in [101], where a simplified physical model is proposed by exploiting first principles and the computation of certain performance indicators, whereby the alarm triggering residual threshold was manually tuned.

However, more recent work primarily considers black-box reference models, of which both linear and non-linear approaches were studied. Xiao et al. [100] deployed the multiple linear regression (MLR) model proposed in [133] in this context while an adaptive estimator served for determining the alarm triggering residual threshold. An MLR based

reference model was also introduced in [24] and compared with a simple linear regression and a decoupling based FDD approach, whereby the latter was found to yield the highest fault detection performance. Another approach is presented by Beghi el al. [102], where various models for statistical analysis of time series data (ARMAX, ARX, etc.) were empirically evaluated and applied for predicting the values of some preselected features.

Other authors, in turn, studied the application of non-linear regression models. Zhao et al. [98], combined support vector regression with exponentially weighted moving average control charts. Similarly, Tran et al. [135] concluded that by replacing the support vector regression model with a radial basis function (RBF) neural network, the overall classification ability can be improved, which was later confirmed within an additional comparison study [136]. Shortly thereafter, the authors proposed a least squared support vector regression model and applied an evolutionary optimisation algorithm for tuning the hyperparameters [99], a term whose meaning will be clarified later.

Knowledge-Based

Although many analytical fault detection models differ greatly in terms of their respective working principle, their approach towards fault isolation appears quite similar because most contributions employ decision tables, which belong to the group of knowledge-based methods. Other works consider the application of Bayesian networks as, for example, proposed by Zhao et al. [101], He et al. [137] or Wang et al. [134] for FDD. However, it seems that in the case of fault detection, contributions that use knowledge-based systems were rarely considered.

Data-Driven

Data-driven approaches are, in turn, widely investigated in this context. Han et al. [74] combined the fault detection and isolation task by employing a support vector machine (SVM) classifier in a multi-class fashion, whereby the classes represented the normal operating states and each investigated fault respectively. The authors further used a genetic algorithm to select significant features from the dataset, which they employed to demonstrate improved classification accuracy. In general, the feature reduction problem has been widely addressed in a great amount of scientific contributions. The reason for this is that the overall performance of the model can be improved if only the most significant features are used to derive a data-driven classifier [6].

One of the most important dimensionality reduction algorithms used for feature selection tasks is principal component analysis (PCA), as it allows the identification of a lower dimensional space of uncorrelated features by decomposing the data into the principal component space, which represents the largest proportion of variance, and the un-modelled residual space [138]. In [139] a SVM classifier for both fault detection and isolation was trained by exploiting the principal component space. Li et al. [87] followed a similar idea by proposing a one-class classifier for detecting fault related anomalies, known as support vector data description (SVDD). However, instead of training the algorithm by use of the principal components, the authors exploited the residual subspace while only observations stemming from the normal chiller operating condition were used to perform PCA. It was found that by considering the residual subspace, most of the system variability caused by changing operating conditions rather than faults is filtered out and the model becomes more sensitive to anomalies [6]. Their model could, thus, outperform the

model proposed by Zhao et al. [138], which was trained on the original input space by exploiting a total number of 16 features that were partly derived from a study published in [140]. Beghi et al. [91] followed a similar approach, but replaced the SVDD classifier with a one-class support vector machine (OC-SVM), originally proposed by Schölkopf et al. [114]. However, both the SVDD and the OC-SVM algorithm are known to yield the same results when *translation invariant kernels* are employed, such as the RBF kernel [141, p. 233]. Two years later, Beghi et al. [6] enhanced their former model by identifying the respective control limits through Hotelling's T^2 statistic and the squared prediction error (SPE), rather than by training an OC-SVM classifier. The major advantage of the model is that it bypasses complex hyperparameter optimisation methods, which usually require that both labelled normal and fault related data are available.

One-class classification (OCC) also gained attention in [142], where a SVDD classifier was trained for each class of both normal and faulty operating conditions respectively. This is particularly advantageous in the later use of the model, as it enables the classification of unknown faults, i.e. the ones not considered during the classifier training phase. Another concept is introduced by Han et al. [132], who used a least squared support vector machine for FDD and demonstrated that it can outperform a SVM as well as a probabilistic neural network. Despite the use of SVM based methods, some authors applied further data-driven concepts in their work, some of which have proven to be at least equivalently effective. One example is demonstrated by Li et al. [121] who applied linear discriminant analysis in this context. Lately, artificial neural networks (ANNs) have come to the fore, as proposed by Yan and Hua [143], who developed a long short-term memory network for chiller fault isolation. Another data-driven concept is the aggregation of several types of algorithms in order to circumvent individual model weaknesses. One such an example, known as ensemble learning, is provided by Zhang et al. [144].

One of the greatest challenges arising from the development of data-driven models constitutes the general lack of appropriate datasets that are needed to successfully train FDD models. This was addressed by, for example, Yan et al. [30], who proposed a generative adversarial network to augment the available labelled training dataset with artificially generated observations and showed that this measure indeed improved the final classification performance of the applied model. Recent work also considered the transferability of FDD models across different chiller types, which is also the subject of this dissertation. One approach is proposed by Fan et al. [29] who applied the synthetic minority oversampling technology on the PCA transformed and reduced dataset, whereas the principal component space was considered. In their work, the authors assumed that labelled data from the source domain, the majority dataset, is more common than from the target domain, to which they refer to as the minority dataset. After oversampling the latter, the trained SVM classifier showed improved classification performance on the combined dataset.

Hybrid

Even though most FDD models proposed during the last decade use either analytical or data-driven models for fault detection, only a few papers also address the application of hybrid approaches. Yan et al. [145] suggested the use of an auto-regressive model with exogenous variables (ARX) to stationarise the time series data before applying a SVM for both fault detection and diagnosis. To this end, a parameter vector was derived from the ARX regression coefficients at each iteration and used as input for a classifier. A

few years later, the authors extended their work in [85], where an online learning fault detection model was proposed by combining an extended kalman filter with a recursively refined OC-SVM.

2.5.2 Summary

From the reviewed papers, a clear trend in the development of CBM approaches towards the use of data-driven models can be deduced. As shown in Figure 2.11, about 50 % applied data-driven methods, while only 11.5 % considered hybrid and 7.7 % knowledge-based approaches. The reason for this may be explained due to the progress in the field of ML in recent years, which allows the use of self-adapting algorithms by simultaneously requiring less a priori information about the observed process [17, p. 33] and no deep understanding of the physics of the system concerned [142]. This is also underlined by the fact that, at 30.8 %, analytical models are still being developed in non-negligible numbers, with most of the work relying on reference models based on either linear or non-linear regression, which can also be attributed to the field of ML. Interestingly, the most common algorithm used for both regression and classification is the SVM or one of its many derivatives, as it is a powerful algorithm for dealing with data characterised by non-linearity and, quite often, high dimensions [139]. It can be concluded that 77 % of all of the above reviewed papers concerning data-driven approaches applied this concept.

Both data-driven and analytical FDD models seem to be highly valued in the research community, and so far promising results have been presented by using these approaches. The development of the latter, however, can be a challenging task due to un-modelled system dynamics, model uncertainty or incomplete knowledge of key parameter values [6]. In addition, the alarm-triggering features and the associated threshold values must be defined, which may require additional effort. Data-driven methods, in turn, can bypass many of these limitations as they enable the use of hidden information buried within the training dataset [121]. Since such models aim to identify patterns from the underlying distribution, they can reliably distinguish between different chiller operating conditions, even if important features are not present [142]. This results in decisive advantages in

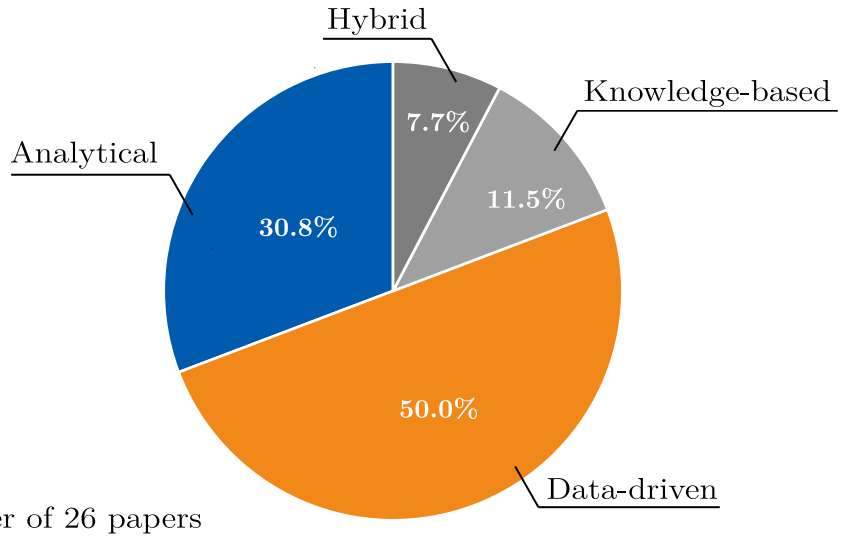

Figure 2.11: An overview of reviewed papers concerning chiller CBM models.

terms of cost efficiency and model accuracy if appropriate datasets can be obtained, as confirmed by the proportion of papers in this area targeting data-driven approaches, which is the motivation behind this dissertation.

2.6 Research Gap and Questions

Although CBM has been subject of research for a long period of time and to this day many papers exist dealing with their application for chillers, there is still a lack of available datasets needed for their development. In particular, data corresponding to different chiller fault types would be of high value [6], but are rarely available, which inhibits the broad application of such methods. In fact, from the total number of 26 papers reviewed in the previous section, only the works carried out in [6], [24], [29] exploited other datasets rather than the one collected by the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) within the project 1043-RP [133], which will be further discussed in Section 4.1. This indicates that the influence of different chiller characteristics on the CBM model performance has rarely been investigated and that most previous works have been shown to be functioning for a dedicated system only. It can thus be assumed that the model development process will have to be repeated for another type of chiller, which implies that both labelled normal data and data related to faults from the respective system are available. However, this assumption does not hold in practice mainly for two reasons: first, most training datasets lack from labelled anomalous data, which curbs the application of common supervised ML methods [6], and second, the CBM model development as well as the data labelling phase are both time-consuming and costly [29], resulting in low economic attractiveness. On the other hand, labelled data associated with a normal operating state of the chiller can easily be obtained from the system [142], e.g. directly after the commissioning phase or after maintenance work has been carried out, whereas most of the available data are often unlabelled. This leads to the assumption that the cost-effective development of data-driven CBM models for chillers can only be archived by using primarily unlabelled data as well as a fraction of labelled data derived from the normal operating condition of the respective system. However, this especially poses greater challenges for the derivation of fault detection and fault isolation algorithms, since the majority of existing models originate from the field of supervised machine learning and thus assume the completeness of all labels within the available dataset. As shown in Table 2.4, there are only few contributions in this context that deal with the development of data-driven models when labels are scarce. Although most existing novelty detection approaches enable the identification of faults by exploiting only labelled normal data in the training phase, they often require labelled

Table 2.4: Overview of papers considering data-driven FDD models.

Task	Available chiller dataset		
	Fully labelled	Partially labelled	Unlabelled
Fault detection (normal data)	[30], [74], [87], [121], [132], [138], [139], [142], [144]	[6], [91]	-
Fault isolation (fault related data)	[30], [74], [121], [132], [139], [142]–[144]	[29]	-

abnormal observations in the model validation phase. The reason for this is that model-specific parameters, also called *hyperparameters* [138], have to be tuned for controlling the learning process by applying, for example, cross-validation [132], [139], [142]. So far, only the work carried out in [91] and [6] particularly addressed this problem by applying heuristic parameter tuning methods.

It is noteworthy that the development of fault detection models in the presence of exclusively unlabelled data has hardly been investigated in the past. Furthermore, the limited availability of labelled fault data becomes an even greater challenge for the fault isolation task, as the automatic assignment of data patterns to specific faults requires the exploitation of the associated class labels. One promising approach to this is proposed by Fan et al. [29] where a knowledge transfer from a fully labelled source dataset and a partially labelled target dataset was realised to train a *cross-domain classifier*. In spite of the fact that their model achieved reliable classification performance on the target chiller dataset, it leaves open the question how this task can be accomplished when labelled, and even unlabelled, fault instances are not present in the target dataset. As a consequence, the aim of this dissertation is to propose a holistic chiller CBM model that enables the cost and time benefits of data-driven approaches to be further exploited. Table 2.5 summarises the research questions and their associated hypotheses in this context.

Table 2.5: Research questions and hypotheses.

Research questions	Hypotheses	Proof
RQ1 - Does heterogeneity between different chillers influence the affect the overall CBM model classification performance?	**H1.1** - fault patterns differ due to unequal operating conditions and design characteristics	4.4
	H1.2 - the fault isolation ability is reduced if the CBM model is applied to different systems without taking into account their heterogeneity	2.4.2
RQ2 - How can partially labelled data from the target system be optimally exploited in the CBM model development process?	**H2.1** - specifically emphasising fault-indicating features improves knowledge transfer between heterogeneous systems and requires only labelled data from the normal operating state	2.4.2
	H2.2 - the use of additional unlabelled data can benefit the development of a classification algorithm	5.3.1
RQ3 - Can costly data labelling be omitted while fulfilling the demands imposed by CBM?	**H3.1** - fault detection can be performed solely depending on the partially labelled data from the target system	5.3.1
	H3.2 - shared fault indicative features exist across domains enabling the CBM model transferability	4.4
RQ4 - How is the CBM model development decoupled from the availability of fault-related data of the target system?	**H4.1** - transfer of knowledge can be achieved through the identification of a shared latent space by use of labelled (normal) data only	2.4.2
	H4.2 - highly fault-indicating features are shared between heterogeneous systems and, therefore, fault patterns of the source system are sufficient to develop the fault isolation algorithm	2.4.2

3 CBM Model

This chapter presents the proposed CBM model including its subcomponents, deriving its architecture from the research questions presented previously. First, the overall architecture design is introduced which is then followed by the description of the principles underlying each concept, from the data filtering techniques used to the fault assessment algorithms located within the FDD block. This section furthermore presents the essential sub-modules from a theoretical standpoint and introduces the mathematical notation in the context of this work.

3.1 Architecture

In the following, the model architecture from the layers defined in Section 2.2.2 is presented by first outlining the problem formulation, which is then followed by the description of the assumptions made. Subsequently, an abstract view of the proposed CBM model is elaborated on in more detail.

3.1.1 Task Formulation

The initial concern driving the proposed CBM model is placed on the acquisition of data, which is processed in a way that fault sensitive features are provided within the scope of the fully automated health assessment task. It starts with the conversion of the raw measurements recorded through a condition monitoring system [17, p. 44], whereby the mapping of measurable to non-measurable state variables as well as the increase of the SNR are the core objectives of the prepossessing and the feature extraction layer. As not all information provided through the CM system is useful in terms of machinery health assessment, one may preselect fault sensitive quantities based on domain knowledge or through adequate feature selection methods to reduce both the number of sensors installed as well as the computational complexity [138]. Accordingly, these two layers can be summarised primarily in finding a mapping of the existing signals onto a feature vector $\grave{x}$ containing useful information about the current machinery health status [58, p. 180]. Yet, further data transformation and mapping may be required throughout the CBM information processing and, thus, feature extraction is not only limited to the initial layers of the model, as will be further emphasises in the remainder of this section.

Although determining an appropriate set of features is indeed important in order to perform condition assessment measures for a dedicated system, most recent research has been devoted to fault detection, fault isolation or even both. This might be explained by the complexity in the development of convenient algorithms induced by, for example, non-linear state variables or limited amounts of available data [134]. However, since these algorithms represent the knowledge about certain faults and their characteristics [58, p. 180], they are at the very heart of the automatic machinery degradation assessment.

One intuitive way to derive an appropriate CBM solution appears to be the training of an arbitrary ML classifier using an existing dataset that contains a sufficient representation of the underlying faults patterns tied to their associated labels. Although this seems to be a straightforward approach, and most works employ such supervised machine learning models [6] in this context, it neglects the heterogeneity amongst the various chiller types. This places a detrimental constraint on the applicability of such models, as

they can be deployed for a single system only. As concluded in the previous sections, the transferability of data-driven models onto a different chiller system has rarely been investigated [29], which implies that most existing models are expected to exhibit degraded fault classification performance when applied to a system other than the one they were trained on. Recall that this is the core of Hypothesis H1.2, which will be verified later in Section 5.3.2. In order to circumvent costly data labelling for each system individually, an alternative to existing approaches must therefore be found. From the hypotheses presented in Section 2.6, it is assumed that this problem can be coped with by utilising only unlabelled and a fraction of normal labelled data stemming from the target chiller, which may be obtained at low cost. In addition, knowledge transfer from a fully labelled source dataset shall be applied within the model training process to meet the requirements imposed by the fault isolation task.

Based on the definitions and notations of [116] and [146], the core problem at hand can be formulated as follows: Let $X_s \in \mathbb{R}^{(n_s \times k)}$ be the available data from the source chiller, with k representing the number of features and n_S the number of observations contained, with the marginal probability distribution $\mathcal{P}(X_s)$ and the feature space $\mathcal{X}_s$. Then $\mathcal{D}_s = \{\mathcal{X}_s, \mathcal{P}(X_s)\}$ can be denoted as the *source domain* with a specific learning task $\mathcal{T}_s = \{\mathcal{Y}_s, \mathcal{P}(Y_s \mid X_s)\}$, $Y_S = \{y_{s_1}, y_{s_2}, \ldots, y_{s_{n_s}}\} \in \mathcal{Y}_s$ representing the labels and $\mathcal{P}(Y_S \mid X_s)$ the conditional source domain probability distribution. Here, the labels can represent either the normal operating condition or a fault considered within the fault isolation phase. Similarly, let $X_t \in \mathbb{R}^{(n_t \times k)}$ be the data sampled from the *target domain*, where n_t represents the number of available observations. The target domain, i.e. the application domain of the CBM model, can be denoted as $\mathcal{D}_T = \{\mathcal{X}_t, \mathcal{P}(X_t)\}$ including its learning task $\mathcal{T}_t = \{\mathcal{Y}_t, \mathcal{P}(Y_t \mid X_t)\}$, where the feature space $\mathcal{X}_t$, the marginal probability distribution $\mathcal{P}(X_t)$, the label space $\mathcal{Y}_t$ and the conditional probability distribution $\mathcal{P}(Y_t \mid X_t)$ are describing the representation of the target system.

Recall that in the case where the conditions $\mathcal{D}_s = \mathcal{D}_t$ and $\mathcal{T}_s = \mathcal{T}_t$ holds, the task becomes a traditional machine learning problem [116] and can thus be solved by training a supervised ML classifier in $\mathcal{D}_s$. However, based on the problem description above, this does not necessarily apply to the detection and isolation of chiller faults, especially when the design characteristics between the source and the target system differ greatly. From this it can be deduced that the primary task is to develop a model $h(x_t)$ that enables the fault detection and fault isolation task to be performed in the target domain. By including the information provided through $\mathcal{D}_s$ and $\mathcal{T}_s$ in the model training process, the model shall additionally avoid costly data labelling to a large extent. To avoid any confusion hereinafter, it should be noted that according to [116], the classifier $h(x_t)$ can be written as $\mathcal{P}(y_t \mid x_t)$, i.e. both notations are equivalent, whereby the former will be used throughout this work.

3.1.2 Assumptions

The key aspects driving the model development process in this work can be reduced to a few core assumptions that significantly determine the concepts and methods to be applicable. Accordingly, from the conclusions described so far, the following assumptions can be reported:

1. **Availability of normal labelled data in the target domain**
   As faults are rare events [6], normal data samples are usually available in the target

domain, which can be obtained e.g. directly after commissioning the target chiller or after maintenance measures have been carried out. As a consequence, this information may be introduced in the model training phase to improve the classification accuracy of $h(x_t)$. Accordingly, one can define the available target domain labels as $Y_t = \{y_{t_1}, y_{t_2}, \ldots, y_{t_{n_t}}\} \in \{-1, 1\}$, whereby $y_{t_i} = -1$ refers to unlabelled and $y_{t_i} = 1$ to normal labelled observations. Similar to the work in [6] and [91], the fault detection task may thus be completed through the use of target domain information only.

2. **Shared feature space**
   Although the presence of different system designs complicates the development of domain insensitive CBM approaches, it is assumed that there exists a shared feature space, which is sensitive to the monitored faults. The assumption therefore is that $\mathcal{X}_s = \mathcal{X}_t$.

3. **Discrepancy between both domains**
   The difference between the distributions is caused by the heterogeneity amongst industrial chiller systems on the basis of their design and operating conditions. As a consequence, it can be assumed that a domain shift is present in the available data and the marginal distributions are misaligned $\mathcal{P}(X_s) \neq \mathcal{P}(X_t)$, as illustrated in Figure 3.1. This will be referred to as *domain discrepancy* in the following.

4. **Fault characteristics are similarly pronounced across domains**
   As faults are detected and isolated through a change within the thermodynamic condition of the target chiller system [133, p. 25] they follow certain patterns caused by physical laws and dependencies. Therefore, it can be assumed that the conditional probability distributions of the source and target domain are strongly related as $\mathcal{P}(Y_s \mid X_s) \approx \mathcal{P}(Y_t \mid X_t)$. Thereby, it should be noted that, according to Csurka [146], the relation $\mathcal{P}(Y_s \mid X_s) = \mathcal{P}(Y_t \mid X_t)$ can be too strong and may not hold in practice.

5. **Shared label space**
   Since the proposed shall benefit from the information provided by the fully labelled source domain, the labels space must similarly be shared. It is consequently deduced that $\mathcal{Y}_t = \mathcal{Y}_s = \mathcal{Y}$. From the opposing point of view, it cannot always be guaranteed whether the target domain fully represents the aimed label space at the model training time and, thus, the model should also be robust in the case $\mathcal{Y}_t \subset \mathcal{Y}_s$.

6. **High process variability**
   Both $\mathcal{D}_s$ and $\mathcal{D}_t$ are subject to a high degree of variability caused by changing operating conditions, as concluded in [6], [87], [91]. This leads to the assumption that by removing the process-related dynamics from the available data, the fault patterns can be distinguished with higher confidence.

7. **Relation between fault severity and statistical distance**
   Following the considerations of [91] or [121], severity level estimation can be performed by introducing a reasonable distance metric as a measuring tool to meet the requirements induced by the fault identification task. This is motivated by the idea that observations associated with low SL are expected to lie closer to the normal data cluster, while high SL are most likely to cause strong differences.

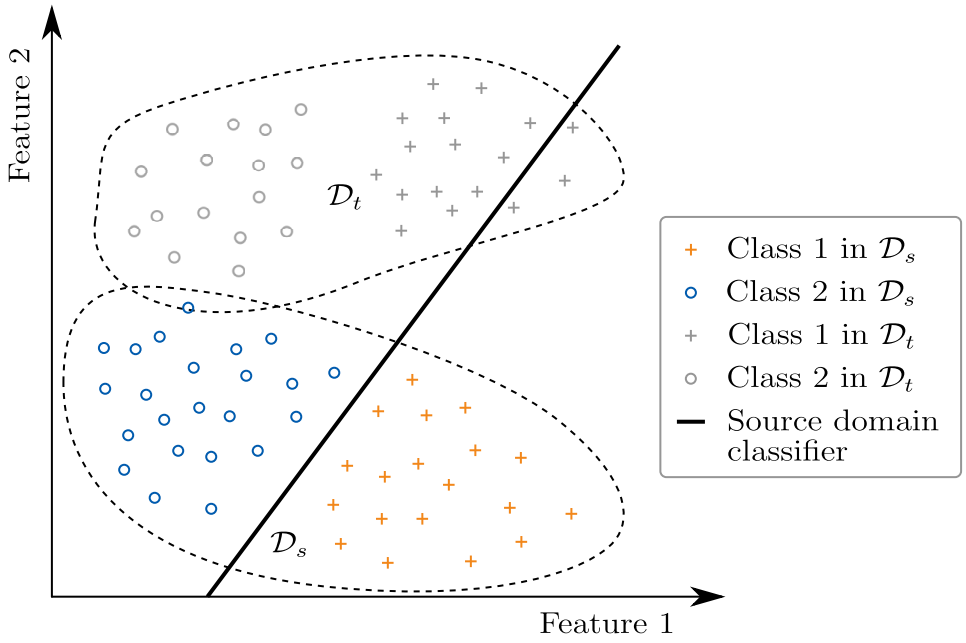

Figure 3.1: Toy example of domain discrepancy between the marginal distributions of two heterogeneous systems as derived from [36].

By concluding the assumptions above, the main objective of the proposed CBM model can be divided in two consecutive tasks, namely: (1) *partially-supervised* [147, pp. 171-208] *fault detection*, for which OCC is employed, and (2) *unsupervised domain adaptation* [36] in the context of fault isolation, which is often also referred to as *cross-domain fault diagnosis* (CDFD) [148]. The fault detection approach should therefore build on the utilisation of normal labelled and unlabelled target domain data. Although this form of learning task is often referred to as semi-supervised ML, it is perhaps not quite correct to introduce this term for the problem at hand, since it assumes the availability of labels from all classes [147, p. 9], albeit in small numbers.

To fulfil the demands induced by the fault isolation task, the model shall enable the transfer of knowlegde from a fully labelled dataset to partially labelled one. Therefore it is considered a domain adaptation problem, which is known to be a subfield of transfer learning where the relations $\mathcal{D}_s \neq \mathcal{D}_t$ and $\mathcal{T}_s = \mathcal{T}_t$ hold [116]. As stated above, it is particularly considered an unsupervised domain adaptation problem. However, by taking into account the precondition that one class of the target system is accurately labelled, namely the one associated to the normal system operating conditions, one may argue that the term unsupervised is not well placed in this respect. Nevertheless, it has to be acknowledged that the other closely related term, semi-supervised domain adaptation, is far less suitable, as no labelled fault patterns are available during the training process. Furthermore, the remainder of the section will show how unsupervised techniques can be employed to reduce the domain discrepancy, which is why the term unsupervised domain adaptation is considered to be applicable throughout this work.

3.1.3 Model Principles

Against this background, a holistic CBM model can be derived to solve the individual problems inferred from the above task formulation with regards to the previously identi-

fied data processing layers. The model is divided into two phases, namely training and deployment. While the former refers to the use of historical data from both the source and target domains, the latter phase designates the continuous observation of online data in the scope of the application, i.e. the data recorded during the model runtime. Through the sequence of data processing layers presented in Section 2.2.2, the system's data are automatically evaluated in such a way that conclusions can be be drawn about the systems degradation state. It also allows information to be derived about any fault condition that may be present while placing fault detection and isolation in the center of attention.

As illustrated in Figure 3.2, steady-state detection and filtering techniques are required within the data preprocessing layer to improve the SNR. This ensures that the informa-

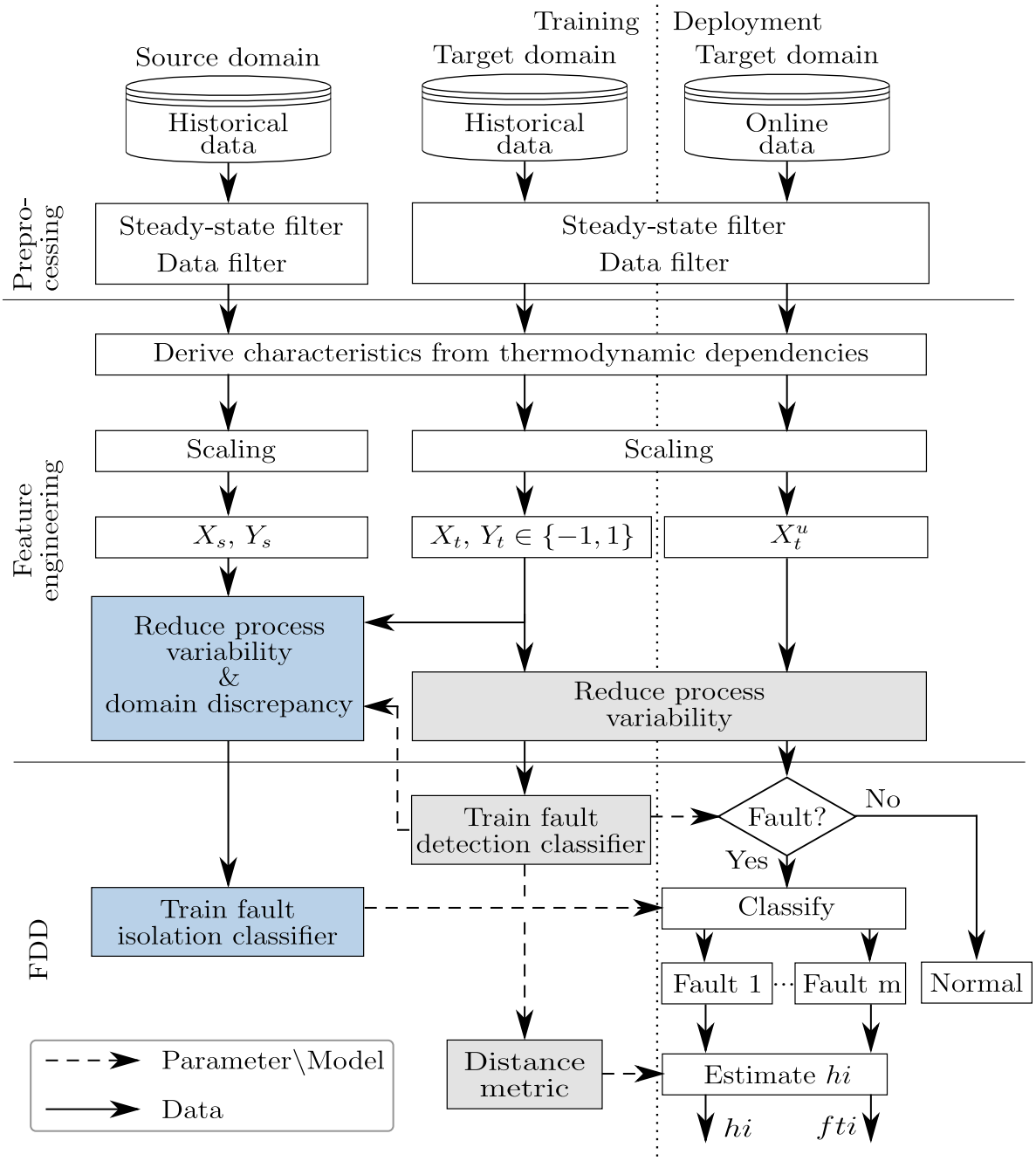

Figure 3.2: Flow chart diagram showing the principles of the proposed CBM model.

tion content provided though the available data is appropriately representing the actual thermodynamic system condition and is less affected by disturbances. Nonetheless, the data filtering process strongly relates to the respective chiller application as well as the respective CM system and may therefore be implemented for each system individually. To extend the available sensory acquired information, feature extraction is performed within the following layer. According to [6] and [133], the determination of new features with high sensitivity to chiller faults is beneficial to the predictive accuracy of the model and is thus an essential factor to be considered. Since it is assumed that the feature space is fully shared between the two domains, it should be noted that the use of same state variables from the source and target systems is necessary under the proposed approach.

The first action that is performed as part of the data transformation layer is known as data scaling. This step is particularly important since the magnitudes amongst the different features can differ greatly due to their units of measurement or simply because of the different physical quantities they represent. Therefore, data scaling is introduced to avoid the domination of high magnitude features, i.e. features representing a wide range of values, during the model training process [81, p. 14] to improve its classification performance [142]. Although the above steps of the data processing scheme are indeed necessary to fulfil the requirements set by CBM, they are common in the context of data-driven techniques and thus do not specifically address the core issues determined from the task formulation presented in Section 3.1.1. However, both fault detection with partially labelled data and CDFD require further feature transformations and selection procedures for mainly two reasons: first, the fault isolation step requires the alignment of the two distributions of $\mathcal{D}_s$ and $\mathcal{D}_t$ to enable the transfer of knowledge from the source to the target domain, and second, both steps rely on features that are less affected by the process variability induced through varying operating conditions. In the final fault evaluation step, the CBM model aims at estimating the health index (hi), representing the fault severity level, for the case when a fault has been detected and isolated. Thereby, a statistical distance metric is introduced using the discrepancy between the anomaly and the normal labelled data cluster as an indicator, which will be discussed in greater detail in Section 3.7. Note that the pseudo-code for the CBM model training is shown in Appendix A.1.2, while the Appendix A.1.3 shows the model deployment.

3.2 Preprocessing

The first action performed in the data preprocessing layer is the separation of data originating from transient system states, such as during start or shut-down phases, or in case of rapidly changing chiller operating conditions [149]. Yet, completely ensuring a steady-state operation of the system does not seem feasible from a practical point of view due to minor disturbances, such as deviations caused by the applied control or as a result of measurement noise. This leads to a criterion that is introduced as an indicator for the automatic rejection of observations that represent transient system states, hereafter referred to as *steady-state detector*.

Although there exists a great amount of proposed steady-state detectors across the literature, such as the work carried out in [150] or [151], this work utilises the approach proposed by Beghi et al. [6] due to its fast convergence capability. It goes without saying that one may employ other techniques towards steady-state detection by considering additional factors like computational complexity or robustness, which is not particularly addressed in this thesis. In general, the technique proposed in [6] relies on computing

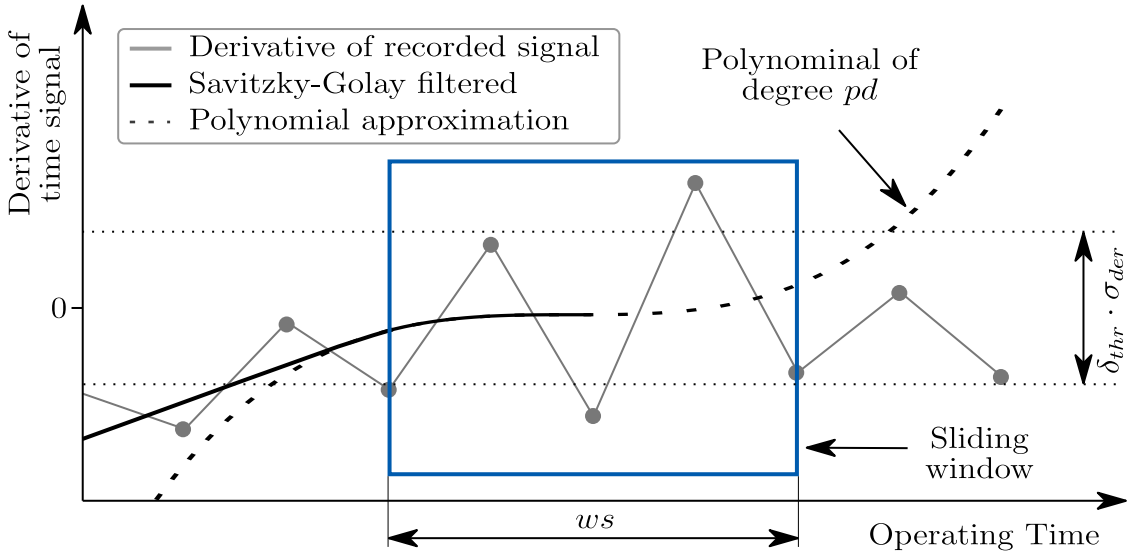

Figure 3.3: Example of Savitzky-Golay smoothing and steady-state detection for $ws = 5$ and $pd = 3$.

the local derivatives of some preselected signals representing the time course of specific thermodynamic state variables using the backward difference approach. The resulting signal is then smoothed through a Savitzky-Golay filter to remove high-frequency noise through least-square polynomial approximation. Then, by setting an appropriate threshold, steady-states are detected when the observed signal does not exceed the upper bound or does not fall below the lower bound for a certain period of time, as exemplary demonstrated in Figure 3.3. As can be seen from this example, the algorithm of [6] builds up on three preselected parameters, namely the sliding window size ws, the polynomial degree of the approximated target function pd and the predefined threshold, which is set $\delta_{thr} \cdot \sigma_{dev}$, whereby σ_{dev} denotes the standard deviation of all computed derivatives and δ_{thr} is a pre-set scalar. For a practical example, the reader is referred to Section 4.3.3 and Figure 4.7 in particular. Furthermore, the pseudo-code for the implementation of the steady-state detector can be found in Appendix A.1.1.

Another concern is placed on random fluctuation of the sensor readings, known as measurement noise. Certainly, the impact on the overall CBM classification performance may arguably be less significant as the integration of a steady-state detector. However, this constitutes a non-negligible factor to be considered to increase the SNR [17, p. 16]. To achieve this, a Savitzky-Golay filter is again applied for smoothing the signals recorded through the CM system. Lastly, the remaining outliers must be removed from the dataset as they have a negative impact on the fault assessment performance [152], for which this work simply applies a set of predefined rules.

3.3 Feature Extraction

The problem of extracting features that are highly sensitive to chiller faults has been discussed in a variety of scientific contributions, but the selected the feature spaces considered in the scope of CBM differ across the literature. For example, Beghi et al. [6] specifically constructed the feature space to be sensitive to the investigated faults by introducing i.a.

the isentropic compressor efficiency or the overall heat loss coefficient. Other authors [85], [138], [149], in turn, followed the results of the study conducted by Comstock et al. [133] as a basis for deriving their feature space, whereby various feature selection techniques were later applied to reduce the amount of features. However, most feature extraction approaches, unlike feature selection, have in common that they essentially rely on domain knowledge to be defined.

The research provided by Comstock et al. [133] also constitutes the basis for the definition of the feature space throughout this dissertation. However, two notes should be outlined here in particular. First, as has been discussed in Section 3.1.1, the shared feature space between the source and the target domain must be employed for the fault isolation relying on domain adaptation. This implies that some features, even though they might be indicative of some faults, are to be discarded from the available dataset unless they represent the same physical quantities across both domains. While this is true for the demands placed on the fault isolation layer, this does not necessarily apply to the other two layers within the FDD block. Secondly, it cannot always be guaranteed that sufficient information is available on how well some features indicate certain faults, and therefore the subsequent feature selection phase should be decoupled from the availability of domain knowledge, which will be addressed in more detail below. It should also be noted that the features considered in this work will be introduced in Section 4 and are furthermore summarised in Appendix A.5.

3.4 Feature Transformation

After defining the set of suitable features, scaling must be performed to reduce the influence of the order of magnitude of certain physical state variables on the respective classification algorithm. In general, two approaches are commonly considered in this context, min-max scaling and *standardisation* [153, p. 37]. Accordingly, min-max scaling scales all features within the dataset in the range [0,1]. However, sometimes this places some impediments in the model training process, as the data might be distorted due to the presence of outliers. As a consequence, most existing work in the field of chiller CBM applies the standardisation technique, as in [74] or [149], which scales the data to zero mean and a standard deviation of one.

Assuming that for each available feature $\grave{x}_j$, regardless of whether it originates from the source or the target domain, its standard deviation σ_j and its mean μ_j can be computed. Then the scaling for each value $\grave{x}_{i,j}$ is defined as [153, p. 37]

$$x_{i,j} = \frac{\grave{x}_{i,j} - \mu_j}{\sigma_j} \tag{1}$$

with $x_{i,j}$ representing all values of the dataset after performing standardisation. At this point, an important remark should be made from a theoretical point of view. As can clearly be seen from the definition of the standardisation scaling approach, the data is assumed to be Gaussian distributed. However, previous studies [138], [154] showed that chiller data actually violate this assumption. In contrast, there exist a large number of scientific contributions in this field, for example [6], [87], [91], [139], that employ techniques that have emerged from dealing with Gaussian distributed data. One of the most common algorithms building upon this is principal component analysis (PCA), which has been shown to yield excellent results in terms of improving the overall fault classification accuracy despite the distributional assumption. From this it can be deduced that

although one has to deal with non-Gaussian distributed data, such methods can still be applied for chiller CBM in practice. This increases the set of possible algorithms to be applied, but also simplifies the way in which the problem at hand can be coped with.

Following the abstract representation of the proposed CBM model shown in Section 3.2, further feature transformation and selection steps are necessary to meet the requirements induced by partially-supervised fault detection and CDFD. Yet, these processes, of which PCA will play a key-role too, can be assigned to one of these two tasks and are therefore described in more detail in the following sections.

3.5 Fault Detection

This section introduces the approach towards the detection of novelties, which are considered to be faults in the context of this work. To this end, an overview of the algorithm is first provided, followed by an explanation regarding the reduction of process variability induced by changing chiller operating conditions that affects the available data. Subsequently, the classification algorithm is introduced that exploits both labelled normal samples and unlabelled target domain samples during training to optimise for a decision boundary that best separates novelties from normal data patterns. The last section then introduces the way the model handles unknown data patterns, i.e. observations that do not originate from the marginal distribution of the target domain at the time of model training.

3.5.1 Overview

According to the previously presented task descriptions, the proposed fault detection algorithm should be insensitive to process variability. This implies that a matrix decomposition method should be introduced to filter out features explaining most system dynamics while maintaining features emphasising the available fault characteristics. As has been shown in previous studies [6], [87], [91], one solution on this is to transform the design matrix using PCA, which is an unsupervised technique that has been widely applied for many data dimensionality reduction tasks. To do so, PCA aims at identifying a lower dimensional feature space that explains most of the data variance [139]. In this context, however, it is adapted to solve for a manifold representing the majority of different chiller operating conditions. As shown in Figure 3.4, PCA is first performed on the normal labelled data to identify the lower dimensional manifold. In the following step, the design matrix is decomposed into the modelled principal component subspace (PCS) and the un-modelled residual component subspace (RCS) [87], the former of which is supposed to represent at least a predetermined cumulative percentage value (CPV) of the variance associated with the normal labelled patterns. As will be described in greater detail in the following section, instead of considering the PCS, the RCS is exploited to train the fault detection classifier because it is expected to be increasingly sensitive to the presence of novelties.

The second task in this context leads to the definition of the type of classification algorithm to be applied, which is known to be a one-class classification (OCC) problem [155]. Recall from the task formulation (1), the problem at hand deals with the learning of a decision boundary to detect novelties by exploiting the available target domain data of both labelled and unlabelled observations and is thus considered to be a partially labelled learning problem. More specifically, a classifier is to be developed from positive,

the normal labelled, observations and unlabelled data, which is commonly referred to as *positive-unlabelled* (PU) *learning* [156, p. 171]. To deal with this kind of problem, this dissertation follows the approach of Liu et al. [157], who proposed the biased support vector machine (BSVM), which will be more accurately elucidated in Section 3.5.3. One of the key benefits arising from this approach is that it allows the hyperparameter of the classifier to be selected in a meta-heuristic manner, such that common parameter tuning methods, e.g. *grid-search* and *cross-validation*, can be applied for this, which otherwise proves to be difficult.

Unfortunately, the BSVM has its limitations when it comes to the detection of novelties associated with unknown data patterns, a problem that is comprehensively explained in Section 3.5.4. Thus, the model is additionally enhanced by a one class support vector machine (OC-SVM). By combining the algorithms above, a novel fault detection algorithm for chillers can be derived, the principles of which are presented hereinafter.

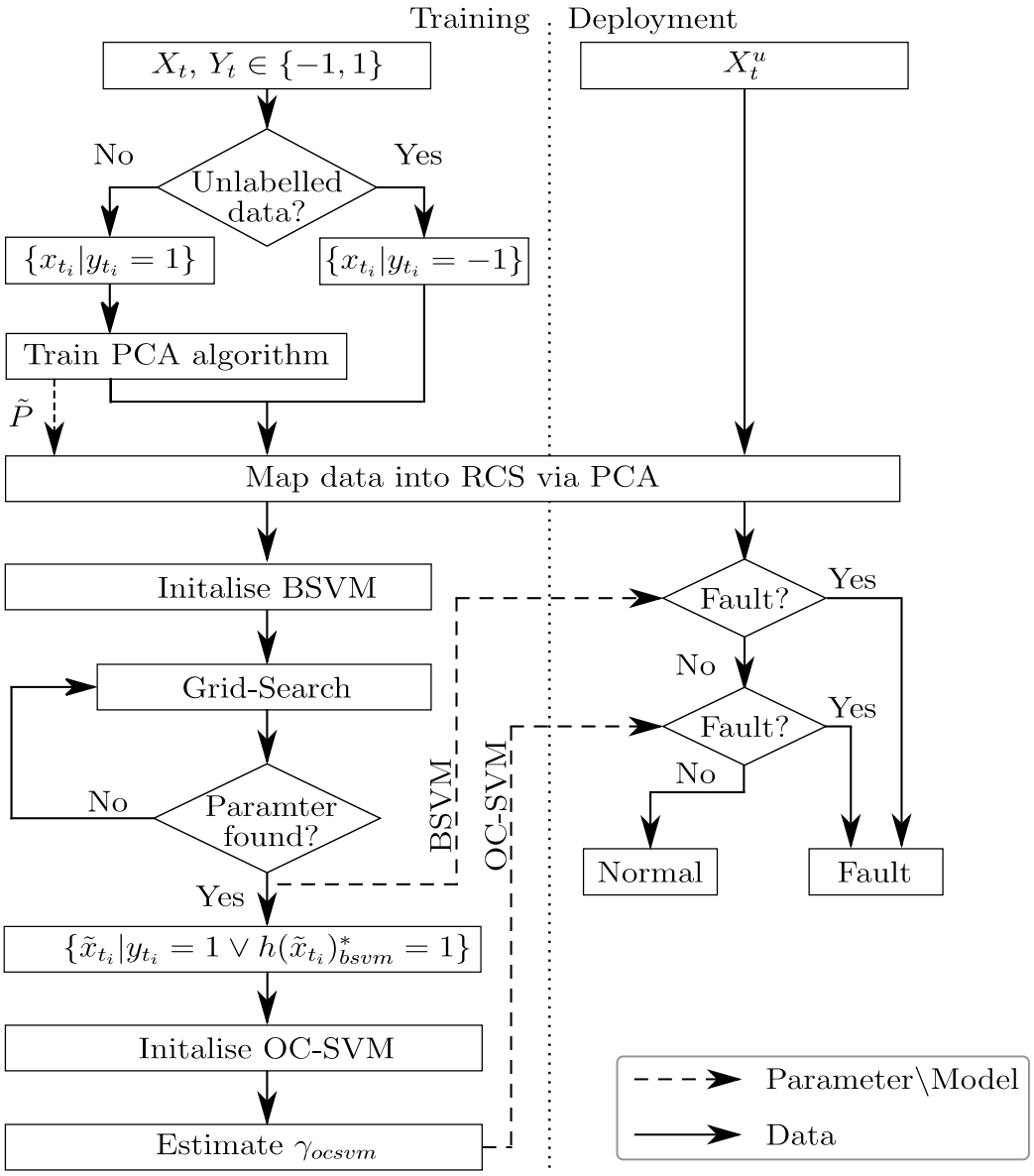

Figure 3.4: Flow chart showing the fault detection algorithm.

3.5.2 Reducing Process Variability

To identify the lower dimensional manifold that models the normal chiller operation, PCA is introduced. In this way, a transformation, or more precisely a rotation, of a matrix into such an axis system is solved so that the largest part of the variance is captured by the first axis, the principal components (PCs) [153, p. 42]. An important aspect of using PCA to reduce the process variability is that is performed by solely utilising the labelled data instances originating from the normal chiller operating condition. Following the notations given above, the resulting matrix can be defined as

$$Z = \{x_{t_i} \in X_t \mid y_{t_i} = 1\} \tag{2}$$

whereby $Z \in \mathbb{R}^{(n_n \times k)}$ denotes the design matrix containing the known n_n fault-free observations sampled from the target domain. It is worth underlining again that any observation associated with $y_{t_i} = -1$ is unlabelled, which can potentially represent a novelty but also a sample belonging to the normal class. In any case, this information is not available. Furthermore, it should be noted that the design matrix Z should be centred according to its mean vector $\mu_n = \{\mu_{n_1}, \mu_{n_2} \ldots, \mu_{n_k}\}$. Then, one can compute the covariance matrix of Z as:

$$\Sigma = \text{cov}(Z) = \frac{1}{n_n - 1} Z^T Z \tag{3}$$

Since Σ is both symmetric and positive semi-definite, one can apply eigendecomposition to diagonalise the representation as [153, p. 42]

$$\Sigma = P \Lambda P^T \tag{4}$$

where the columns of P, also known as the loading matrix, represent the linear independent eigenvectors of Σ, while Λ is a diagonal matrix containing the associated eigenvalues. It should be noted that one can alternatively apply singular value decomposition directly on the design matrix Z to perform PCA, which is also the general framework in this respect [153, pp. 42]. Following the notation given in [6], the design matrix Z can consequently be decomposed as

$$Z = \hat{Z} + \tilde{Z} = \hat{T}\hat{P}^T + \tilde{T}\tilde{P}^T \tag{5}$$

where the columns of $\hat{P} \in \mathbb{R}^{(k \times k_{pc})}$ are the first k_{pc}, a hyperparameter whose meaning will be especially discussed later, eigenvectors corresponding to the highest eigenvalues of Λ spanning the PCS. Furthermore, $\hat{T} \in \mathbb{R}^{(n_n \times k_{pc})}$ is the associated principal score matrix. Conversely, $\tilde{P} \in \mathbb{R}^{(k \times (k - k_{pc}))}$ represents the unmodelled RCS, while $\tilde{T} \in \mathbb{R}^{(n_n \times (k - k_{pc}))}$ is the residual score matrix.

As shown in Figure 3.5, the intuition behind this approach is to identify the subspace from Z which explains most system variability. Since only known normal samples are selected for performing PCA, the direction of the PCs are excepted to model a large part of the various chiller operating conditions. From the opposing point of view, the residual components (RCs) essentially represent both measurement noise and, most importantly, novelties. Thus, by mapping the target domain data onto the un-modelled RCS space, the overall fault sensitivity may significantly be improved, which benefits the subsequent classification process.

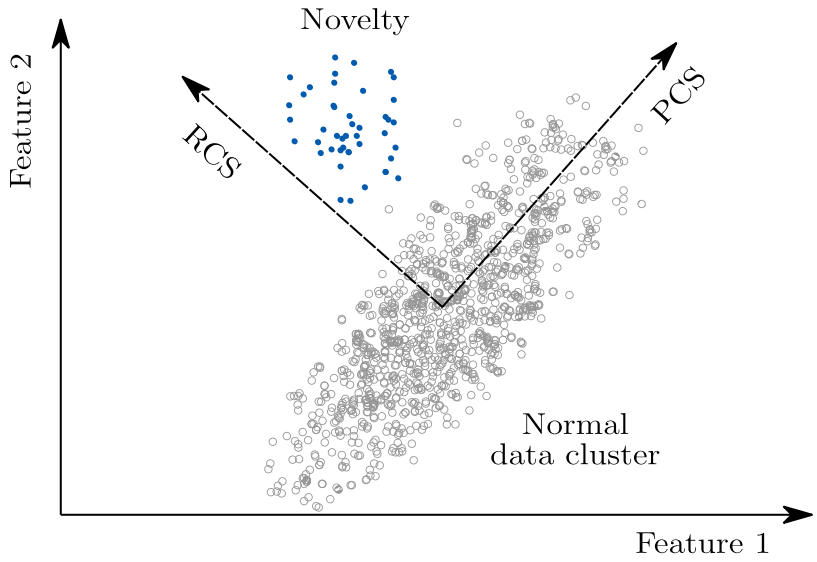

Figure 3.5: Example of matrix decomposition into principal component and residual component subspaces showing the effect of the reduction of process variability contained within the data.

Accordingly, the decomposition of any target domain sample into the two orthogonal subspaces can be defined as [6]

$$x_{t_i} = \hat{P}\hat{P}^T x_{t_i} + \tilde{P}\tilde{P}^T x_{t_i} \tag{6}$$

whereby $\hat{P}\hat{P}^T x_{t_i}$ is the modelled and $\tilde{P}\tilde{P}^T x_{t_i}$ is the unmodelled part of the target domain observation. Accordingly, the data dimensionality can be reduced by mapping any observation to its residual components as:

$$\tilde{x}_{t_i} = \tilde{P}^T x_{t_i} \tag{7}$$

It goes without saying that this equally applies to the unknown observations $x^u_{t_i}$ in the model deployment phase, while both are centred according to μ_n. With the data representation of $\tilde{x}_{t_i}$ exploited in place of the original input x_{t_i}, a classification algorithm can be trained to solve for an appropriate decision boundary to separate novelties from normal data.

However, one challenge arises from the determination of k_{pc}, e.g. the number of discarded PCs, which is a trade-off in terms of improving the overall fault classification performance. On the one hand, a low number of discarded PCs preserves much information content, but the data may still be affected by high system dynamics. On the other hand, a high value of k_{pc} can also lead to the reduction of the fault sensitivity, as significant features are removed. One intuitive approach towards the determination of k_{pc} is to relate it to the cumulative percent value (CPV) of the explained variance represented by the columns of $\hat{P}$, as described in [87] or [139]. According to Han et al. [139], the number of discarded PCs can then be computed by means of the eigenvalues of Λ, which, in turn, represent the variance along the axis of the PCA transformed coordinate system, as

$$CPV(j) = \frac{\sum_{i=1}^{j} \Lambda_{i,i}}{tr(\Lambda)} \cdot 100\ \% \tag{8}$$

with $\Lambda_{i,i}$ representing the eigenvalues along the diagonal of Λ in descending order. By setting a predefined threshold $CPV_{k_{pc}}$, k_{pc} can be computed as:

$$k_{pc} = \min\left(\{\ j \in \{1, 2, \ldots, k\} \mid CPV(j) \geq CPV_{k_{pc}}\ \}\right) \tag{9}$$

Yet, a reasonable value for $CPV_{k_{pc}}$ must be chosen, which is not uniformly defined in the literature. For example, in [91] the minimum CPV was chosen to be greater than 99.3 % while in [6] it was 95 %. Other contributions, in turn, indicate $CPV_{k_{pc}} = 85$ % to be an appropriate threshold [158], [159]. It is obvious that a correct parameter setting is needed for the model to converge to an appropriate solution and therefore a sensitivity analysis with regard to this parameter will be performed and presented in Section 5.3.

It becomes evident from the definitions above that by finding a linear transformation of the design matrix Z through PCA, most system variability can be captured by the PCs. By decomposing Z into its PCS and RCS, the latter in particular can be used to derive the classification algorithm. This way, the model becomes more sensitive to novelties and the overall fault detection performance is expected to be increased.

3.5.3 Learning from Positive and Unlabelled Observations

So far, several data processing steps have been introduced, ranging from data filtering techniques to reducing the process variability contained in the available data of the target domain, all serving one goal: improving the overall classification performance of the proposed model. This still leaves open the question how a classifier can be derived in this regard. Referring to Hypothesis H2.2 presented in Section 2.6, exploiting labelled as well as unlabelled target domain samples can benefit the fault detection process, as the latter may carry important information about certain fault characteristics, even though their actual labels are not available. Since this particular problem addresses the case of a binary classification, i.e. indicating a normal or abnormal chiller operating condition, it can be ascribed to a special subfield of ML, which is commonly known as PU-learning. In fact, this differs from most fault detection algorithms applied in this field of research in that it does not attempt to perform *density estimation* [160], but rather aims at modelling the class probability distribution with respect to the binary classification problem.

As was concluded in Section 2.5.2, most data-driven approaches in the field of CBM employ variations of the SVM classification algorithm, as it comprises decent capabilities in terms of dealing with problems affected by non-linearity, high dimensions, local minima as well as the availability of only small amounts of observations while utilising principles originating from the statistical learning theory [139], [161, pp. 7-60], [162]. The algorithm itself is a binary classifier, but can also be extended to operate in a multi-class fashion, which will be introduced in Section 3.6.3. In general, the concept behind SVMs is to find a separating hyperplane between two sets of samples, in which a margin between them is maximised.

As shown in Figure 3.6, the algorithm aims at identifying the support vectors (SV) that are critical for the given classification task. There are two cases to consider: In the first scenario, there exists a hyperplane that best separates the samples of two classes and the algorithm can solve for what is known as a hard margin hyperplane. In practice, however, this does not always hold, e.g. due to the presence of outliers, and often one has to deal with problems in which the patterns are not linearly separable. Therefore, allowing the algorithm to violate this assumption [141, pp. 203-205] to a certain degree

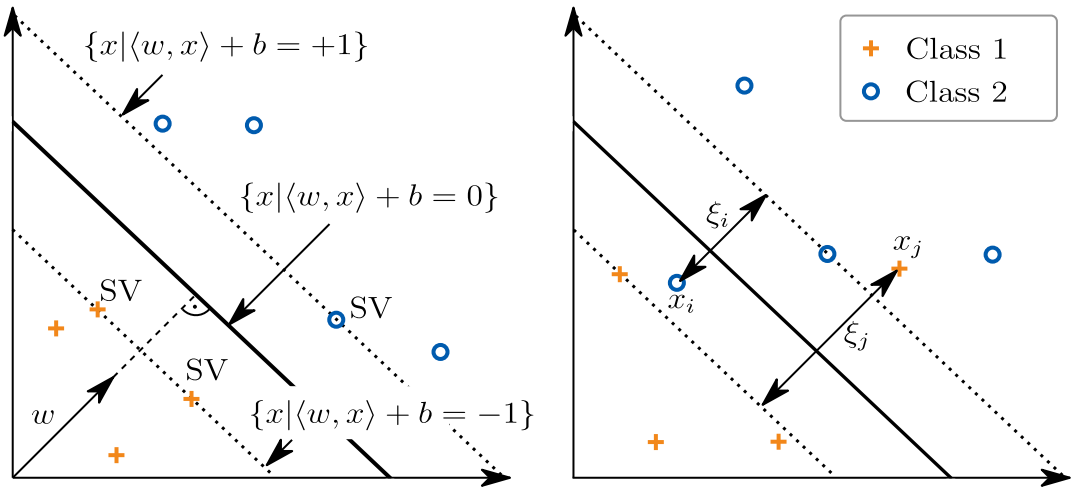

Figure 3.6: SVM classification as hard margin hyperplane (left) and soft margin hyperplane (right), as partially derived from [161, pp. 10-16] and [141, p. 191].

can be of great value. This is the idea behind soft margin hyperplanes. The primal form for an arbitrary two class optimisation problem can then be defined as

$$\begin{aligned} \underset{w,b,\xi}{\text{minimise}} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i\left(w^T x_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned} \tag{10}$$

where w is a vector orthogonal to the hyperplane, C is a regularisation parameter for handling the trade-off between minimising the margin error and maximising the margin and ξ_i a slack variable that is $\xi_i > 0$ for observations located on the wrong side of the hyperplane or the ones that lie within the margin [141, p. 206]. Furthermore, b is a bias term and $y_i \in \{-1, 1\}$ is the actual label of x_i.

The definition of (10) applies in the case when the data is fully labelled and is considered the ground truth in such as scenario. It is worth noting that the above optimisation problem will play a further role in the context of fault isolation and is therefore referenced again in Section 3.6. However, for the problem at hand, where the majority of observations remain unlabelled, the standard soft margin SVM does not seem to lead to a suitable solution. To cope with this, this work follows a certain variation of the SVM proposed by Liu et al. [157], known as the biased support vector machine (BSVM). Although its principles are not much deviated from the one given above, it allows to deal with the case of PU-learning by weighing positive errors, i.e. the ones induced by observations from the normal class $y_{t_i} = 1$, and those provoked from unlabelled data $y_{t_i} = -1$ differently. This is achieved by assigning different regularisation parameters in the two cases during the optimisation process. Following the notations introduced previously for the target domain, the primal optimisation problem can hence be defined as

$$\begin{aligned} \underset{w,b,\xi}{\text{minimise}} \quad & \frac{1}{2}\|w\|^2 + C^+\sum_{i=1}^{n_n}\xi_i \; + C^-\sum_{i=n_n+1}^{n_t}\xi_i \\ \text{subject to} \quad & y_{t_i}\left(w^T \tilde{x}_{t_i} + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned} \tag{11}$$

where C^+ and C^- are the regularisation parameters assigned to the positive and unlabelled patterns respectively. Thereby, one should note that instead of solving (11) using the original input data representation of the target domain, it is performed on its RCS by exploiting $\tilde{x}_{t_i}$. Unfortunately, the primal form is commonly not the optimal approach to deal with and (11) restricts itself to solve for a linear decision boundary only. A more suitable way towards solving the problem at hand might be to look at its Lagrangian dual form instead, which can be defined as

$$\begin{aligned}\mathcal{L}(w,\xi,b,\alpha,\beta) = \frac{1}{2}\|w\|^2 + C^+ \sum_{i=1}^{n_n} \xi_i \; + C^- \sum_{i=n_n+1}^{n_t} \xi_i \\ - \sum_{i=1}^{n_t} \alpha_i \left(y_i \left(w^T \tilde{x}_{t_i} + b\right) - 1 + \xi_i\right) \\ - \sum_{i=1}^{n_t} \beta_i \xi_i\end{aligned} \tag{12}$$

where α_i and β_i are the Lagrangian coefficients, also known as dual variables. Since (12) must be minimised with respect to w, ξ, b, and maximised by means of the Lagrangian coefficients, the primal variables must first be differently expressed by first forming the corresponding partial derivatives [141, p. 207], which leads to the following conditions:

$$w = \sum_{i=1}^{n_t} \alpha_i y_{t_i} \tilde{x}_{t_i} \tag{13}$$

$$\alpha_i + \beta_i = C^+, \quad \forall i \in [1, n_n] \tag{14}$$

$$\alpha_i + \beta_i = C^-, \quad \forall i \in [n_n + 1, n_t] \tag{15}$$

$$\sum_{i=1}^{n_t} \alpha_i y_{t_i} = 0 \tag{16}$$

Then, by substituting (13)-(15) into (12), one can derive the quadratic optimisation problem of the BSVM algorithm. Besides minimising the number of variables to optimise for, the main advantage of dealing with the Langragian dual form instead of the primal form arises from the inclusion of the dot product of the observations $\tilde{x}_{t_i}^T \tilde{x}_{t_j}$. This circumstance allows this term to be replaced by a kernel function $k(\cdot)$, for which the problem can now be defined as

$$\begin{aligned}\underset{\alpha}{\text{maximise}} \quad & \sum_{i=1}^{n_t} \alpha_i - \frac{1}{2} \sum_{i \neq j \wedge i,j=1}^{n_t} \alpha_i \alpha_j y_{t_i} y_{t_j} k\left(\tilde{x}_{t_i}, \tilde{x}_{t_j}\right) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C^+, \quad \forall i \in [1, n_n] \\ & 0 \leq \alpha_i \leq C^-, \quad \forall i \in [n_n + 1, n_t] \\ & \sum_{i=1}^{n_t} \alpha_i y_{t_i} = 0\end{aligned} \tag{17}$$

where the constraints on α_i already imply the different treatment of positive and unlabelled patterns. Since (17) enables the use of kernels, the algorithm can now be extended

to problems affected by non-linear dependencies. As shown in Figure 3.7, the idea of applying kernels in the context of SVMs is to map the original input space into a, possibly, higher-dimensional feature space through $\Phi(\cdot)$, where a separating hyperplane can now be constructed [141, p. 15].

There exist a variety of kernel functions, such as the polynomial or sigmoid kernel, which all have to satisfy Mercer's theorem to be considered as such. Accordingly, the resulting kernel matrix K, which is also a Gram matrix, must be positive semidefinite [153, p. 323]. Although many different kernels are described in the literature that satisfy this condition, most contributions [74], [85], [91], [132], [138] in the field of chiller CBM have shown that the fault detection performance can be increased by utilising the *Gaussian radial basis function* (RBF) kernel, which is therefore introduced in this work as well. The RBF kernel function is defined as

$$k\left(\tilde{x}_{t_i}, \tilde{x}_{t_j}\right) = \left\langle \Phi\left(\tilde{x}_{t_i}\right), \Phi\left(\tilde{x}_{t_j}\right) \right\rangle = exp\left(-\gamma \|\tilde{x}_{t_i} - \tilde{x}_{t_j}\|^2\right) \tag{18}$$

where γ is the kernel bandwidth coefficient. Furthermore, it follows that (17) and (18) show that non-linear problems can be treated without ever explicitly computing the mapping function $\Phi(\cdot)$, but considering the dot product in the feature space, which is considered a reproducing kernel Hilbert space (RKHS) [141, pp. 35-39] and may span an infinite number of dimensions, as in the case of the RBF kernel [153, p. 324]. This is commonly referred to as the *kernel trick* [141, p. 34], as it avoids expensive, and sometimes even infeasible, computations to be performed. From this it follows that a decision function for any unknown target domain sample $\tilde{x}_t^u$ towards the fault detection task can be defined as

$$h(\tilde{x}_t^u)^*_{bsvm} = \sum_{i=1}^{n_t} \alpha_i y_i k\left(\tilde{x}_t^u, \tilde{x}_{t_i}\right) + b \tag{19}$$

whereby the threshold parameter b can simply be determined from the first constraint introduced in (11) and the condition (13), whereby the SVs with non-zero slack variables should be taken into account for the approximation, i.e. the ones associated with Lagrangian coefficients that satisfy $0 > \alpha_i > C^+$ or $0 > \alpha_i > C^-$ respectively. As a

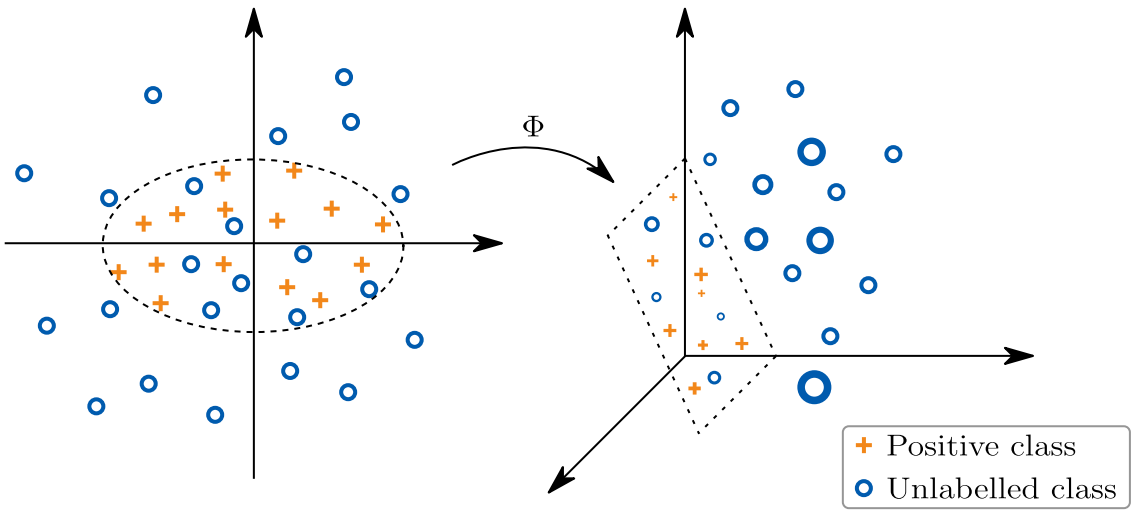

Figure 3.7: Example of a 2D problem mapped into a 3D feature space, based on [141, p. 29].

consequence, (19) relates to the distance of an arbitrary observation vector $\tilde{x}_t^u$ to the separating hyperplane within the feature space and will be referred to again later in Section 3.7. The actual classifier, however, is defined as $h(\tilde{x}_t^u)_{bsvm} = \text{sgn}(h(\tilde{x}_t^u)_{bsvm}^*)$. One further important aspect should be outlined in this regard. Since the dual variables α_i are non-zero for SVs only, the prediction time reduces accordingly, a property that is equal for all SVM based classifiers considered in this work. Moreover, since its Lagrangian dual form differs from that of the standard SVM only by a slightly different constraint, it has an equal time complexity of training time.

It appears that the above formulations lead to a solution of a PU classifier to be applied for chiller fault detection tasks. Yet, the question of how to determine the hyperparameters just presented remains open. This problem is also known as the *bias-variance dilemma* in classical statistics [141, pp. 126] and affects the final classification performance to a large extent. For example, the kernel bandwidth parameter γ must be chosen wisely, since a high value of γ_{bsvm} can allow the model to adapt to a more complex decision boundary, but can also lead to *overfitting* [153, p. 324]. Similarly, C^+ and C^- must be determined as they represent the trade-off between margin errors attributed to positive or unlabelled samples and therefore additionally affect the overall fault detection performance.

In supervised ML, the selection of a set of appropriate hyperparameters is straightforward as it allows to apply empirical risk minimisation [141, pp. 126] by employing, for example, cross-validation [163] and suitable search strategies, such as grid-search, to tune them with respect to the learning problem at hand. Unfortunately, PU learning adds some complexity in this regard because only one class is assumed to be accurately labelled, whereas other labels remain unknown. To cope with this kind of problem, Liu et al. [157] proposed a modified metric related to the well known F-score metric as a performance measure. According to the authors, the modified metric can be defined as

$$\text{F-MOD} = \frac{r^2}{Pr\left[h\left(\tilde{X}_t\right)_{bsvm} = 1\right]} \tag{20}$$

where r is the *recall* and $Pr[h(\tilde{X}_t)_{bsvm} = 1]$ the probability of a sample being classified as normal. As a result, (20) can be employed as a performance measure indicating a suitable set of hyperparameters in the context of the respective search strategy, for which this work applies cross-validation and grid-search in the following.

3.5.4 Handling Unknown Patterns

The BSVM algorithm has been demonstrated to perform well in classification and appears to obtain better results than pure density estimation [155], [157] as it can, at least partially, recover the faults discriminative structure of the target data. This leads to a problem in the deployment phase, namely when emerging novelties lead to unknown data patterns, as in the case of a fault occurring for the first time within a dedicated chiller system for example. In such a scenario, the BSVM algorithm alone is likely to be prone to type II errors, also known as false negatives (see Section 5.2.2 for more detailed explanations), as no associated support vectors are available to recover a suitable decision boundary.

To cope with this, the fault detection model presented in this section additionally comprises a one class classifier relying on density estimation of the available normal data of the target domain. While this covers the case where the discriminative structure between the fault-free operational states and the states associated with anomalies cannot

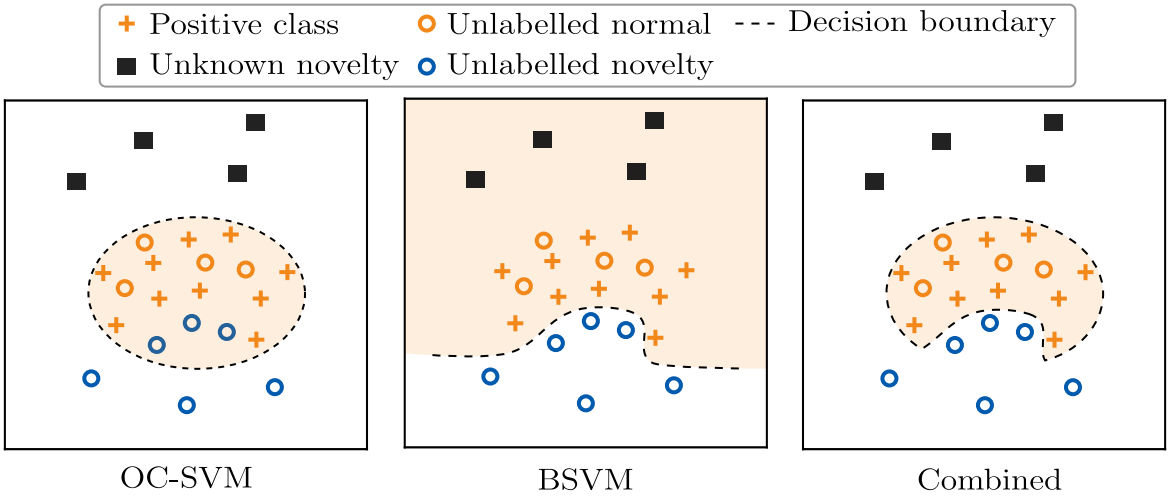

Figure 3.8: Toy example of the combination of density estimation and PU learning.

be recovered from the distribution of the target domain at training time, it may comes at the cost of lowered classification performance. A corresponding experimental proof with regards to this application domain will be presented in Section 5.3.1. There are mainly two OCC algorithms that have emerged from the principles of SVMs and are based on the principles of density estimation that have been successfully applied to the detection of chiller faults, namely the SVDD [87], [138], [142] and the OC-SVM [91], [113] classifiers. The difference between the two is that Tax and Duin [164] constructed the former such that all target vectors are enclosed by a hypersphere with minimal radius, whereas Schölkopf et al. [114] developed the latter as a hyperplane that best separates them from the rest of the feature space with maximal distance from the origin [155]. Interestingly though, if both algorithms are applied with translation invariant kernels, as in the case of (18), they both converge to equivalent solutions [141, p. 234]. In this dissertation, the OC-SVM algorithm is employed, the fundamentals of which are described in [114].

Figure 3.8 illustrates the advantages of combining PU-learning and density estimation in terms of the expected fault detection performance. The concept underlying the proposed approach is based on the results of a multitude of scientific reports [157], [165], [166] indicating the superiority of PU-learning over density estimation in various application domains. Although they tend to model the underlying conditional distribution well, they may not be fully adequate for fault detection purposes, since not all patterns might be available within the training dataset. To overcome this problem, the proposed model combines the two methodologies to achieve both improved classification accuracy for fault patterns buried within the training data and the ability to recognise emerging novelties. Similar to the challenge posed by the estimation of the hyperparameters in the previous section, the OC-SVM relies on the two corresponding parameters ν, the trade-off parameter, and γ_{ocsvm}, the RBF kernel bandwidth. To tune these parameters in an efficient manner, this dissertation follows the suggestions of [91] as follows: Since ν aims at characterising he number of outliers and SVs [114], it is usually set to a small value indicating that the majority of observations from the training dataset are contiguously enclosed. In addition, γ_{ocsvm} must be estimated for which the tightness detection algorithm [167] is applied.

3.6 Fault Isolation

Accurately distinguishing between normal system operations and the ones incurred through faults constitutes an important step towards CBM and is indeed the prerequisite to determine a fault type and to estimate its SL. To accomplish the former, further steps are to be taken because labels associated with faults are not expected to be available from the target domain. One solution to this is to exploit the labelled patterns of a different but related domain to enable fault isolation to be performed, which is described hereinafter. The first part of this section elucidates the principles of the proposed fault isolation algorithm. The domain adaptation method is then presented, followed by the approach towards fault classification.

3.6.1 Overview

The domain discrepancy between the source and the target domain is assumed to be a significant impact factor onto the overall fault classification performance, as stated in Hypothesis H1.2. This is often because some features do not adequately represent the similarities between the two datasets and thus inhibit the application of a fault isolation model to different heterogeneous chiller systems. For some examples, the reader is referred to Section 4.4. In fact, if the domain discrepancy is too large, the model may have a high bias in the target domain [168, p. 23], which must be avoided.

To solve such a problem in a way that the available, and fully labelled, source domain dataset (X_s and Y_s) can be exploited, the similarities between the datasets must first be emphasised more strongly. This leads to a problem known as domain adaptation [169], a special case of transfer learning [146], as has been already described in Section 3.1.1. Although there are multiple methods in this regard, which can be categorised into instance-based, model-parameter-based, relational-based and feature-based transfer learning [168, pp. 10-11], [116], the latter will be applied in this work. In general, the principles underlying this approach are based on the learning of *domain-invariant* features, which generalise well on the target domain [170]. The aim of this measure is to minimise the domain discrepancy caused by the different system-specific characteristics in order to reduce the model's generalisation error.

Although up to this point it can be assumed that most faults can be correctly identified by the fault detection algorithm presented in the previous sections, the actual classes associated with such novelties have so far been disregarded. Accordingly, this dissertation proposes a simple yet effective domain adaptation method based on subspace identification via PCA, whereby both the domain discrepancy as well as the process variability are reduced simultaneously. Due to the same challenge identified as part of the fault detection layer, namely the presence of chiller operating dynamics in the data, this measure allows improving the overall fault sensitivity of the resulting model by decomposing the design matrix through the use of PCA. Subsequently, domain-invariant features are identified by mapping the source domain information into the RCS. In the following step, a classifier is derived from the transformed source domain dataset, with the remaining tasks following common supervised ML procedures.

Since fault isolation aims at identifying the nature of abnormal system behaviour from a variety of possible fault types, it is considered a so-called *multi-class classification problem*. There are a variety of different algorithms that can be employed for this type of problem. Yet, this work applies the concept of SVMs, but extends it so that it is

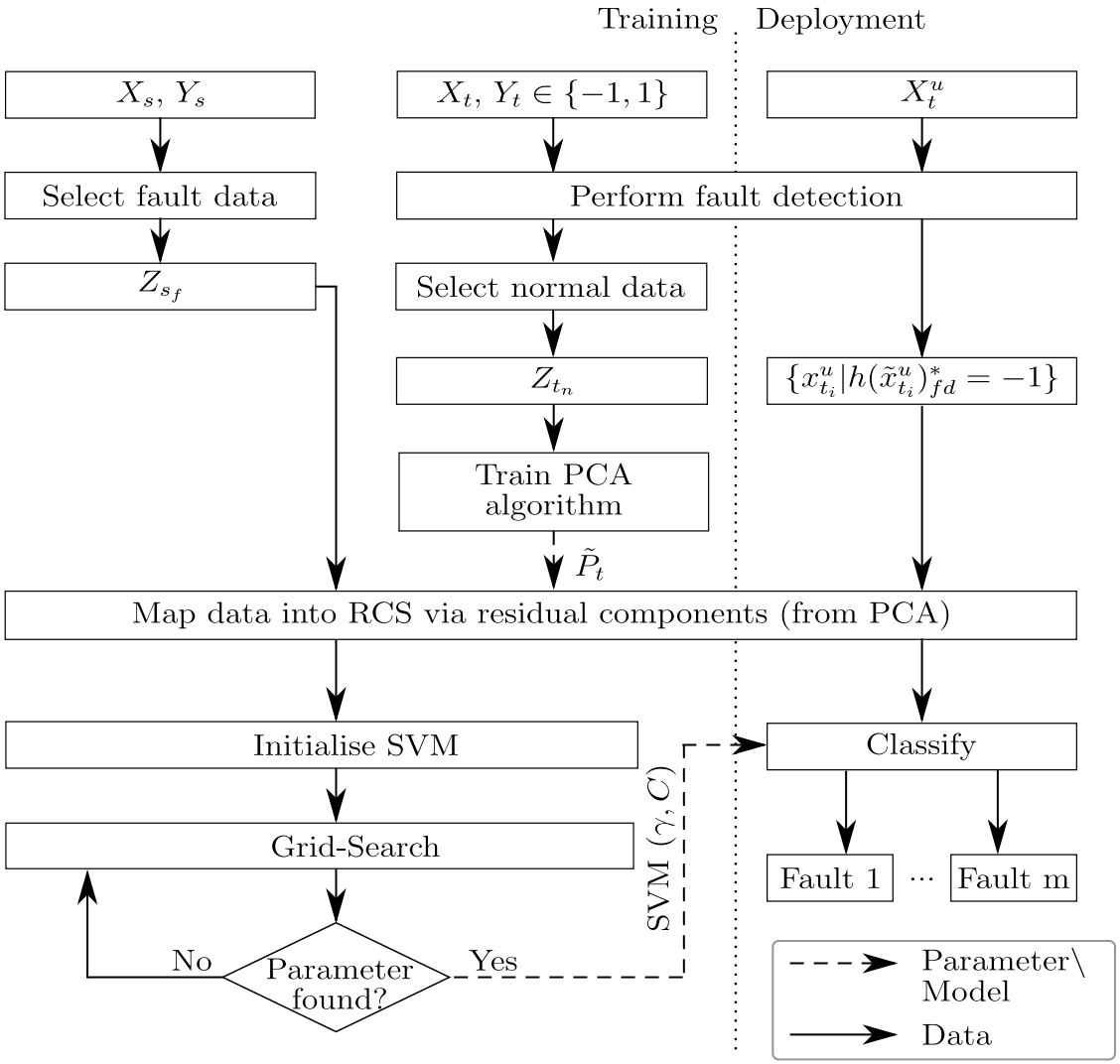

Figure 3.9: Flow chart showing the fault isolation algorithm.

not limited to a binary classification problem. Even though a large number of scientific contributions indicate its superior properties compared to other ML algorithms, the SVM is also subject to the problem of overfitting, an obstacle that must be overcome. Just as in the case of the fault detection algorithm, the problem of finding appropriate and problem-specific hyperparameters arises. However, after reducing the domain discrepancy between both distributions, the classifier is trained solely on the labelled source domain samples allowing the benefits of common techniques, such as cross-validation and grid-search, to be exploited.

3.6.2 Domain Adaptation

Much work has been attributed to the field of chiller fault isolation, in which the authors have achieved promising classification results by applying various supervised ML techniques, mostly based on SVMs. Even though one intuitive attempt is to simply train the classifier on the input space originating from $\mathcal{D}_s$, it may not be an adequate choice because the domain discrepancy may cause the classifier to misinterpret the patterns contained in

$\mathcal{D}_t$. Nonetheless, if one can find domain-invariant features that well capture similarities between the datasets, a cross-domain classifier can be derived yielding decent fault isolation results in the target system. Note that since the feature space $\mathcal{X}_s = \mathcal{X}_t$ as well as the label space $\mathcal{Y}_s = \mathcal{Y}_t$ are expected to be the same and only the marginal distributions differ from one another $\mathcal{D}_s \neq \mathcal{D}_s$, the problem at hand is considered a homogeneous domain adaptation problem [146]. Yet, the case $\mathcal{Y}_t \subset \mathcal{Y}_s$ at the model's training time cannot be neglected in the problem at hand, which therefore differs from many existing domain adaptation approaches, a problem that will be elucidated in more detail below.

One chiller CDFD model has been proposed by Fan et al. [29], whereby a fault classification accuracy of 96.7 % could be achieved for $\mathcal{T}_t$. However, their model relies on a few correctly labelled fault instances from each observed fault class in the target domain to enable the adaptation process, which is not usually available or at least not easily retrievable with little cost and time effort. Consequently, the problem at hand focusses on the more general case, namely unsupervised domain adaptation [171], [172].

In particular, models based on ANNs have been proposed in recent years that allow to align the two given distributions in such a way that the fault discriminative structure of the target domain is well recovered from the source data, whereby promising classification results have been demonstrated in a variety of applications. For example, some researchers aimed at minimising the maximum mean discrepancy (MMD), a discrepancy metric proposed by Gretton et al. [173], between the domains by use of convolutional neural networks [171], [174]. Other authors, in turn, followed the idea of adversarial learning instead [175], [176] that originates from the idea of generative adversarial neural networks [177]. Albeit much research has been devoted to the application of ANNs for domain adaptation purposes, they also bring some limitations that should not be disregarded. The first crucial aspect constitutes the estimation of a reasonable set of hyperparameters [178], since the majority of classification metrics rely on the availability of labelled observations from the target domain. Secondly, deep architectures in particular are notoriously computationally intensive [179] and often require large amounts of data to be trained on, which may not be available.

Apart from the ANN based methods towards unsupervised domain adaptation, there are also other well-known methods for this task, such as transfer component analysis (TCA) [180], correlation alignment (CORAL) [181] or PCA-based subspace alignment (SA) [182], the latter of which is closely related to the proposed approach. Note that these domain adaptation algorithms will be compared to the presented approach in Section 5.3.2.

There are certainly more methods to align the source and target domain distributions for the purpose of fault isolation. However, many of these algorithms are subject to an assumption that is difficult to be satisfied under real conditions. In fact, they assume the occurrence of the observed fault patterns in both the source and target domain dataset. Even though the availability of target domain labels is not a prerequisite for the application of common unsupervised domain adaptation techniques, observing whether or not certain fault patterns are contained in the target data seems to be impractical from an economical perspective and diminishes the added value of data-driven approaches for CBM. Thus, this work proposes an alternative method to reduce the domain discrepancy for chiller fault isolation via the identification of domain-invariant features through PCA, the principles of which are introduced subsequently.

The idea essentially builds on the considerations presented in [6], [87], [91], where better fault detection performance could be achieved by mapping data into the RCS.

Table 3.1: Methods for PCA based identification of transferable features for the presented domain adaptation problem.

Designation	Description	Comments
PCA(X_s), PCA(X_t)	Perform PCA on each dataset separately	Features are mixed up and the residual components retrieved from $\mathcal{D}_s$ may not be well representative in $\mathcal{D}_t$
PCA(X_s)	Map target domain data to the source domain RCS	The resulting feature space might be skewed towards $\mathcal{D}_s$ and is only conditionally applicable for the target system
PCA(X_s, X_t)	Apply PCA on the combined dataset	The feature space results in a compromise between the distributions of $\mathcal{D}_s$ and $\mathcal{D}_t$ but may, however, be less representative for the target chiller
PCA(X_t)	Map source domain data to the target domain RCS	The process variability is reduced by solely concerning the target domain data and a decent feature representation is retrieved for $\mathcal{D}_t$

In fact, if the fault discriminative structure occurs in a highly distinguishable pattern through the identification of the residual components, as described in Section 3.5.2, it may also be well representative across a multitude of related domains. Thus, applying PCA to reduce the process variability and to emphasise a transferable fault structure could constitute a straightforward approach towards CDFD for chillers and, possibly, other sub modules within industrial refrigeration systems. However, the question of how to appropriately apply this method to retrieve a cross-domain fault isolation algorithm arises. There generally exist four approaches to achieve this, which are listed in Table 3.1 by way of comparison. As a consequence, mapping the source domain data to the target domain RCS is conspicuously the better choice and is thus the method considered below. An experimental proof concerning this assumption will be presented in Section 5.3.2. The algorithm can be described as follows:

Let $Z_{ns} \in \mathbb{R}^{(n_{ns} \times k)}$ be the design matrix containing n_{ns} observations from the normal operating condition of the source dataset. Similarly, the observations from the target system are represented by the matrix

$$Z_{nt} = \{x_{t_i} \in X_t \mid y_{t_i} = 1 \vee h(\tilde{x}_{t_i})_{fd} = 1\} \tag{21}$$

where $Z_{nt} \in \mathbb{R}^{(n_{nt} \times k)}$ contains n_{nt} samples of labelled data as well as the observations assigned to the normal class via the previously introduced fault detection algorithm. To reduce the process variability in the following step, Z_{nt} can be decomposed into its PCS and RCS through PCA in a similar fashion as within the fault detection layer by use of the relations (4)-(6), from which the residual space loading matrix $\tilde{P}_t \in \mathbb{R}^{((k-k_{pc}) \times k)}$ can be obtained. For consistency, k_{pc} is equivalently chosen as in the fault detection layer to split the two subspaces within the fault isolation layer. A sensitivity analysis in this regard will also be presented in Section 5.3.2. Moreover, let the set of labelled fault samples of $\mathcal{D}_s$ be

$$Z_{fs} = \{x_{s_i} \in X_s \mid x_{s_i} \notin Z_{ns}\} \tag{22}$$

with $Z_{fs} \in \mathbb{R}^{((n_s - n_{ns}) \times k)}$. From this it follows the mapping of source domain observations into the target domain RCS as

$$\tilde{Z}_{fs} = Z_{fs} \tilde{P}_t^T \tag{23}$$

where $\tilde{Z}_{fs} \in \mathbb{R}^{((n_s - n_{ns}) \times (k - k_{pc}))}$, including the associated labels, now represents the training dataset for deriving a cross-domain classifier as exemplary shown in Figure 3.10. Note that (23) simultaneously applies to any unlabelled $f : x_t \mapsto \tilde{z}_t$ or unknown target domain observation $f : x_t^u \mapsto \tilde{z}_t^u$. As can be seen from the definitions above, the proposed approach performs an asymmetric feature transformation of the source data towards the target domain. Thus, it differs from many other approaches, such as TCA, which try to find a latent space of domain shared features by minimising the MMD distance. According to Sun et al. [181], such domain alignment procedures are often superior to their symmetric counterparts in terms of performance and flexibility, a statement which is also confirmed by other studies [183], [184].

However, one should bear in mind that, although reducing the domain discrepancy is the primary objective of this measure, one crucial factor distinguishes the proposed approach from many CDFD methods. Since most models assume all classes to be fully covered by both domains during the adaptation process, they primarily aim at aligning the fault patterns of the two distributions. Unfortunately, the assumption $\mathcal{Y}_s = \mathcal{Y}_t$ may be too strong in practice, as the presence of certain fault patterns within the unlabelled data of the target domain can only be assessed to a limited extent. Since this is usually observed by humans during the model development process, the cost benefit arising from the data-driven fault isolation procedure remains questionable in this scenario.

Recall from the research question RQ4 and the assumptions presented in Section 3.1.2, that the case $\mathcal{Y}_t \subset \mathcal{Y}_s$ must additionally be taken into account. Even more, this must also cover the worst case, i.e. the unavailability of any fault classes at the time of training. Either way, it must be assumed that this information cannot be obtained, at least not at reasonable cost. In conclusion, the proposed approach towards fault isolation raises promising expectations because;

1. it does not require any target domain fault patterns to be available to converge to a suitable solution and therefore circumvents time-consuming data analysis procedures,

2. there is no hyperparameter estimation required, except for k_{pc}, as in the case of TCA, for example, and is therefore more practical since one can bypass complex parameter search methods that prove to be difficult in unsupervised domain adaptation problems

3. and it is a comparatively simple solution that can be practically implemented with only a few lines of code and may therefore be well suitable for practitioners.

An important aspect of the proposed approach is that the remaining development of the fault isolation classifier can be treated as a supervised ML problem. Thus, a classification algorithm can be derived by exploiting common methods, metrics and optimisation approaches with the only difference being the training on the transformed source dataset, as explained in more detail below. In summary, this leads to decisive advantages in terms of robustness and flexibility, as the fault isolation model can be obtained independently of the availability of particular anomalies in the target dataset. Yet, it must be underlined

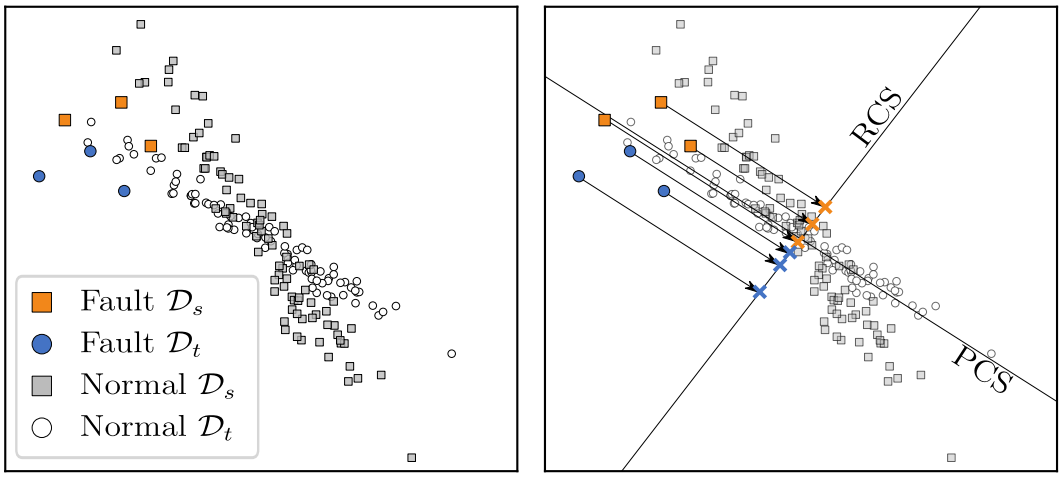

Figure 3.10: Toy example illustrating the mapping of observations from both domains into the target domain residual component subspace.

that the presented fault isolation model may only be applicable for minor domain shifts. In any case, as refrigeration systems are subject to the same thermodynamic principles, the occurrence of related fault patterns are expected to be on a comparable scale.

3.6.3 Multi-Class Classification

In contrast to the fault detection problem, which constitutes a binary decision problem, fault isolation mostly deals with the distinction of more than two fault classes. The number of recognisable classes depends on the available label space $\mathcal{Y}_s$ and may contain a great variety of different fault types to be distinguished. Of course, the number of classes is practically limited due to some constraints, such as richness of data to enable the accurate distinction between the various patterns or simply the costs associated with the labelling process. In any case, the obstacles imposed by the fault isolation step generally lead to a problem known as *multi-class classification*.

Overall, it is barely possible to make a statement about the general classification performance of a particular algorithm due to the myriad of existing classifiers that can potentially be used for such tasks. Thus, the selection of a particular method is difficult and often follows a trial and error approach. Fortunately, the latest research results in the field of chiller CBM, indicate the use of SVMs to be a viable choice due to their superior properties. Originally developed for binary classification problems, they can also be extended to multi-class classification problems, which is the approach chosen for fault isolation in this work. Schölkopf et al. [141, pp. 211-214] describe multiple methods in this regard, of which the "one versus rest" and "one versus one" approach belong to the most common ones. Yet, as shown in previous studies [74], [185], [186], the latter especially shows slightly better performance and is therefore chosen hereinafter.

As illustrated in Figure 3.11, the idea is to define a set of binary classifiers, one for each possible pair of fault types, assigning any arbitrary observation to either of the two respective classes [141, p. 212]. Consequently, a dataset consisting of m fault classes observed within the fault isolation layer results in $(m-1)m/2$ binary classifiers to be trained. The actual class is then determined via a voting mechanism that is employed to estimate the probability of a particular class affiliation. To explain this principle,

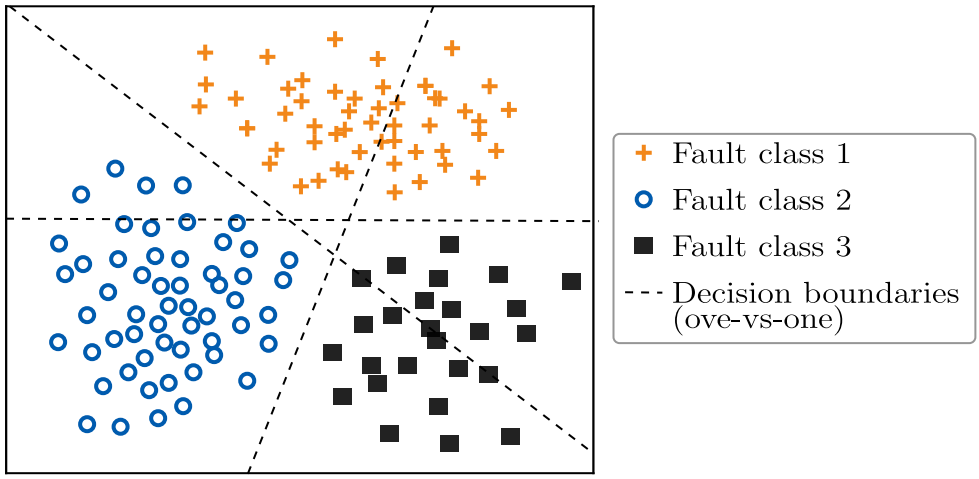

Figure 3.11: Example of a multi-class problem for a linearly separable case.

consider any unlabelled target domain observation $\tilde{z}_t^u$ mapped to the RCS using the relation (23). Let $h(\cdot)_{i,j}$ be the classifier for each possible pair of classes, where $i \neq j$ and $h(\cdot)_{i,j} = -h(\cdot)_{j,i}$. The Lagrangian dual form of the SVM decision function originating from the optimisation problem presented previously in (10) for any binary class combination with $n_{s_{i,j}}$ samples being assigned to either class can then be defined as

$$h(\tilde{z}_t^u)_{i,j}^* = \sum_{k=1}^{n_{s_{i,j}}} y_{s_{i,j,k}} \alpha_{i,j,k} k\left(\tilde{z}_t^u, \tilde{z}_{fs_{i,j,k}}\right) + b_{i,j} \tag{24}$$

with $\alpha_{i,j,k}$ being the dual coefficients, $y_{s_{i,j,k}}$ the labels associated to the source domain observations $\tilde{z}_{fs_{i,j,k}}$ in the respective binary classification context and $b_{i,j}$ representing the bias term of the respective classifier. To determine the class with the highest probability of a correct assignment, a voting system can be introduced, where the class with the maximum count of votes wins as [187], [188]

$$\underset{i}{\operatorname{argmax}} \sum_{i \neq j, j=1}^{m} f(h(\tilde{z}_t^u)_{i,j}), \quad \forall i \in \{1, 2, \ldots, m\} \tag{25}$$

with

$$f(h(\tilde{z}_t^u)_{i,j}) = \begin{cases} 1, & \text{if } h(\tilde{z}_t^u)_{i,j} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

where $h(\tilde{z}_t^u)_{i,j} = \operatorname{sgn}(h(\tilde{z}_t^u)_{i,j}^*)$, as mentioned previously, and $f(h(\tilde{z}_t^u)_{i,j})$ denotes a scoring function. Since this approach may lead to a case where multiple classes receive the same numbers of votes, it might be advisable to additionally include the decision-functions in the decision-making process [187] in such scenarios. To this end, the class with the highest probability can be selected, which is related to the distance of any observation from the separating hyperplanes in the feature space. In the following, the function representing this decision-making process will be referred to as $h(\tilde{z}_t^u)_{fi}$. It is worth noting that unlike the fault detection phase, where the class associated with the normal operating condition of the chiller is primarily considered, in this layer only data associated with faults are

considered in the classifier training process. This is due to the fact that the fault isolation algorithm distinguishes between the different fault patterns, without trying to model the fault-free operational state. It is also worth mentioning that both the RBF as well as the linear kernel function will be comparatively implemented in the validation phase presented later in Section 5.3.2.

3.7 Fault Identification

The two previous layers of the CBM model are clearly the essential key elements proposed in this work. Nonetheless, following the conclusions drawn from Section 2.2.2 as well as the demands presented within the DIN ISO 13374 [64], only detecting and isolating emerging faults may not be sufficient to fulfil the requirements imposed by PHM, as this does not provide any indication about a fault severity. Therefore, the fault identification layer aims at estimating a health indicator representing the severity level. Despite the fact that this is important for estimating the RUL, it should be noted that further activities related to PHM are beyond the scope of this dissertation.

One effective way of performing this task has been proposed by Li et al. [87] and Beghi et al. [91], who both argued that the distance between the fault data cluster and the normal cluster increases with higher SLs. This is further emphasised by the fact that the RCS is particularly sensitive to novelties, as exemplified earlier in Figure 3.5. The introduction of a suitable statistical distance metric that allows conclusions to be drawn about the extent to which a fault cluster differs from a normal cluster therefore seems to be a viable way forward. In fact, this information is already available and has been introduced in Section 3.5 along with the SVM decision function. As has been demonstrated in [87], [91], observations located close to a classifier's decision boundary are likely to represent a fault at low severity, whereas higher distances are considered to be associated with high SL, which is illustrated in Figure 3.12.

So far, several SVM-based classifiers have been introduced in the context of the proposed CBM model, each performing a different task. This, however, leaves open the question which of the resulting decision boundaries should be taken into consideration for severity estimation. The logical consequence though leads to the fault detection model, as it already aims at identifying faults by observing the distance of unknown target domain

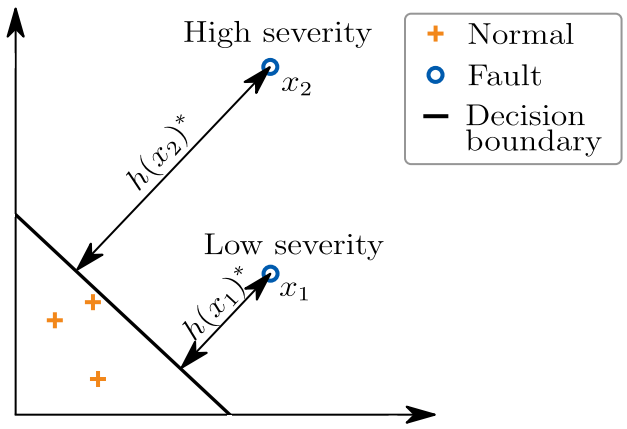

Figure 3.12: Severity level estimation based on distance from decision boundary in the feature space.

samples to the normal data cluster in the feature space. Recall that two classifiers were introduced for this task, of which the aim of the BSVM algorithm is placed on recovering a discriminative structure from normal and unlabelled data, whereas the OC-SVM deals with the case of detecting unknown patterns during the deployment phase. In general, both classifiers are theoretically eligible for severity estimation by employing their respective decision functions as statistical distance metrics. Yet, one obstacle arises from the demands placed by the ISO 13374 [64], which describes the outcome of the CBM evaluation phase to be a health index (hi) ranging from 0 to 10, whereby 0 refers to a complete failure and 10 to a fully functioning system. Following the ideas described above, one intuitive way to achieve this might be to scale the outcome of the classifier's decision function accordingly. Since novelties are characterised by the outcome of negative values of the BSVM or OC-SVM, scaling the negative range seems a reasonable approach, whereby $h(\cdot)^* = 0$ represents the upper bound of the decision function.

Unfortunately, the lower bound of the decision function may not appear that clear at first, since it can theoretically take on any negative value. Yet, by paying attention to the SVM principles, this value can also be obtained. For example, consider the OC-SVM decision function, which is defined as [141, p. 232]

$$h(\tilde{x}_t^u)^*_{ocsvm} = \sum_{i=1}^{n_t} \alpha_i k\left(\tilde{x}_{t_i}, \tilde{x}_t^u\right) - \rho \tag{27}$$

where ρ is the hyperplane offset. Assuming the kernel function $k(\cdot)$ to be the RBF, then one can observe that the first term approaches zero for high distance between the SVs and the unknown observation $\tilde{x}_t^u$ in the feature space, resulting in the lower bound being $-\rho$. Unfortunately, the BSVM differs in this regard, as (19) can theoretically yield lower values as obtained by the bias term and some values can fall below the alleged lower bound. However, the bias term may still be a good indicator to approximate the lower bound, as will be demonstrated later in Section 5.3.3. Accordingly, the negative range of (19) and (27) can be scaled to the desired magnitude to meet the requirements placed on hi.

4 Data Source

The two datasets exploited to validate the proposed model, one of which has been extensively used in most scientific papers on the subject since the late 1990s, are introduced in this section. As concluded in Section 2.5.2, this area of research lacks from experimental studies of chiller faults. Therefore, suitable datasets for the development of data-driven CBM models are scarcely available, an issue addressed in this dissertation. The following sections therefore provide an overview of the available datasets, focusing on the experimental investigation of typical chiller faults carried out as part of this work. First the experimental set-up is presented, including a description of the test rig used as well as the data acquisition procedure, followed by the experimental design. Subsequently, the results are depicted and summarised, with particular emphasis on the individual fault characteristics.

4.1 Available Chiller Datasets

Two chiller datasets containing data from both normal operation and multiple faults are used as the basis for the validation procedure. On the one hand, this allows the model's generalisation ability to be assessed and, on the other hand, the presented approach requires labelled source domain as well as partially labelled (minor amount of labelled samples from the normal operating condition) target domain data as a prerequisite.

The first dataset originates from the experimental investigation of chiller faults conducted by Comstock et al. [133] within the ASHRAE financed project 1043-RP. By adapting a 316 kW centrifugal chiller in their test rig, different operating conditions could be simulated, whereas the cooling water temperature at the condenser inlet T_{ci}, the chilled water temperature at the evaporator outlet T_{eo} and the evaporator heat flow Q_E were used as control variables. For the experiments, the authors identified some predefined set-points from the control variables resulting in a test sequence consisting of a total number of 27 system states. During their experiments, each of these states was approached one after another, while the test sequence was kept unchanged throughout the study. In addition, they introduced a gradient-based steady-state criterion calculated for a 10 minute time window, which had to be met at least once by each operating state before moving to the next state. After performing a series of benchmark tests to reliably identify the normal operating chiller condition, the authors artificially induced several faults, which have been presented in Section 2.4.2, at varying severity levels (SL) and examined their impact on the system variables. For all performed test series, the data were recorded at 10 sec intervals, taking into account primarily analogue sensors, binary states, and thermodynamic variables computed via the programmable logic controller (PLC). To date, the dataset collected as part of this research project is the most widely used in the recent literature [6] and still represents a key-role in the development of modern CBM models.

As shown in Table 4.1, the second dataset stems from the experiments conduced by use of a 100 kW screw chiller in the scope of this dissertation. Similar to the project founded by the ASHRAE, three control variables as well as an associated test sequence were defined to study the effect of certain faults on the overall operational condition. To achieve comparability between the two research projects, some of the faults considered in [133] were also examined as part of this work.

As stated in Hypothesis H3.2, this work assumes the existence of similarities of fault indicative features across heterogeneous chiller systems in order to enable the use of

Table 4.1: Properties of the considered chiller systems.

Property/Component	Centrifugal chiller	Screw chiller
Compressor type	Centrifugal	Screw
Specified cooling load	316 kW	100 kW
Refrigerant	R-134a	R-717
Heat exchanger type	Shell-tube	Plate
Secondary-coolant	Water	Water-glycol
Evaporator design	Dry	Flooded

domain adaptation techniques for fault isolation purposes. Moreover, since there are only few studies on the experimental investigation of chiller faults and their effects on overall chiller operating performance, this work is intended to contribute to the research in this area. Therefore, detailed information about the experiments conducted on an adapted screw chiller test rig is provided in the following sections. Note that the data for the CBM model validation was partially obtained from these experiments.

4.2 Experimental Setup

In the following, an overview of the experimental investigation is provided, which is based on the adaptation of an ammonia-based screw chiller system. First, the test rig is presented including its concept, its function-preserving components as well as its final design. Then, the data collection procedure is described in more detail and the available sensor suite is presented.

4.2.1 Test Rig

The test rig utilised throughout the experiments primarily consists of an ammonia based (R-717) chiller system adapted to enable the simulation of some predefined faults. The main components of the chiller include a screw compressor driven by a three-phase synchronous motor with an electrical rated output of 35 kW controlled via a variable-frequency drive (VFD). Plate heat exchangers are used for both the condenser and the evaporator, the latter being a flooded type. A horizontal vapour-liquid separator, also known as *receiver*, is integrated to separate the liquid and gaseous phases of the working fluid. This prevents the compressor from drawing liquid refrigerant, which could potentially lead to its destruction. In addition, the circulating refrigerant passes through an expansion valve between the condenser and the receiver before entering the evaporator.

One of the most crucial aspect of setting up a test rig for experimentally investigating chiller faults is the wide variety of different operational states that can possibly occur in practice due to external influences. Comstock et al. [133] narrowed down three key influencing quantities impacting the chiller operation, namely: the ambient temperature, the provided cooling load and the cooling temperature. The test rig design must therefore allow for the investigation of different operating conditions in order to study the effects of faults under a wide range of operating conditions. In addition, these conditions must meet sufficiently accurate repeatability to ensure comparable operating states. To meet these requirements in a cost effective manner, two bypasses are introduced: one between

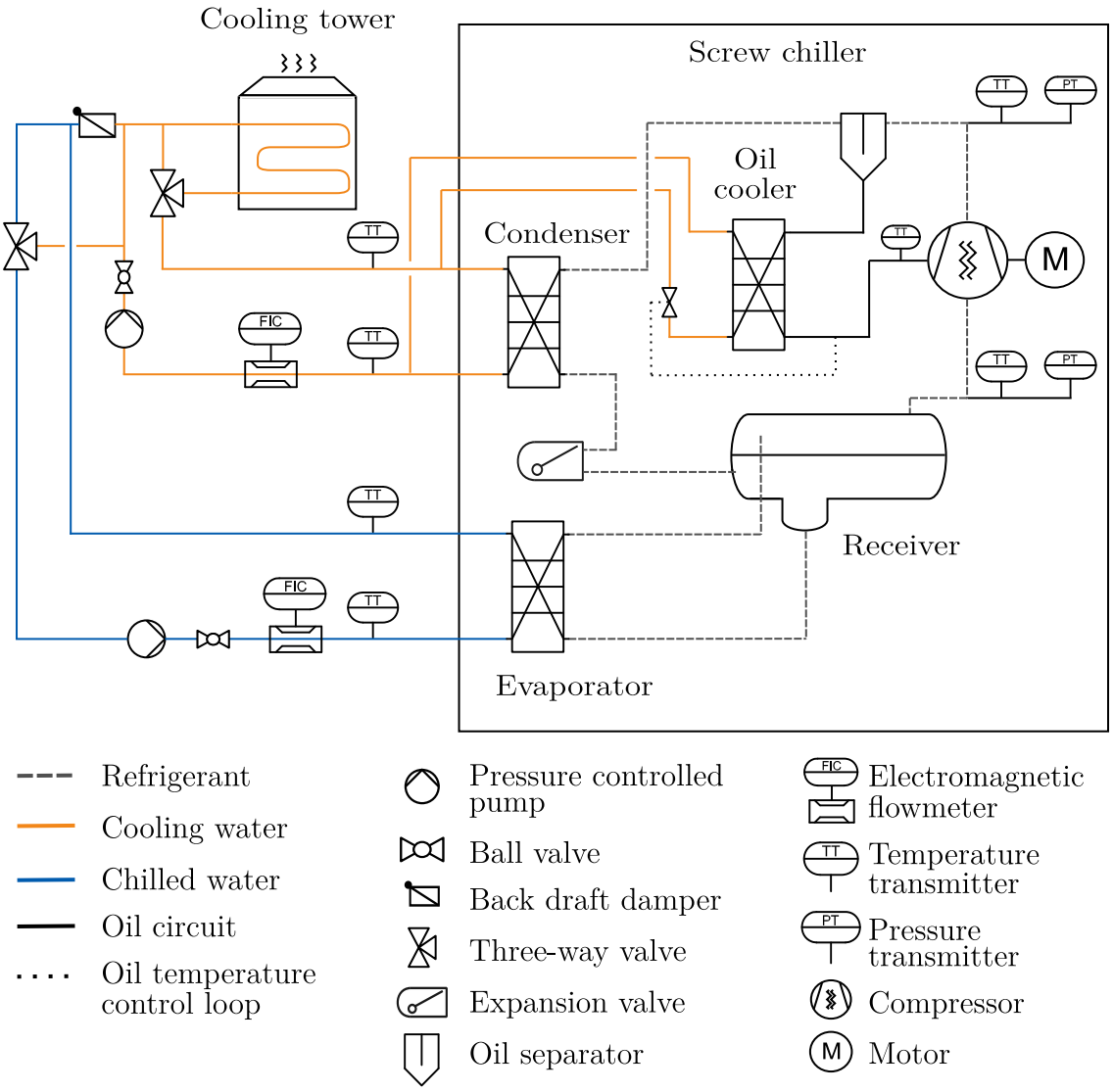

Figure 4.1: An abstract view of the simplified test rig piping diagram.

chilled water circuit and the cooling water circuit, and another between the cooling water circuit and the cooling tower circuit.

As shown in Figure 4.1, the two three-way valves allow to control the water-glycol mass flows entering each circuit, which in turn enables the inlet temperatures to the condenser and evaporator to be set. The excess heat is furthermore released into the external environment via a cooling tower located outside the test facility. Hence, the test rig allows for the simulation of different ambient temperatures that would affect T_{ci} the most. Furthermore, the water-glycol entering temperature at the evaporator inlet T_{ei} can be controlled.

At this point, it should be pointed out that there is a significant difference between real-world refrigeration systems and the test rig presented in this work. In practice, the control strategy of a refrigeration system includes several other subcomponents in addition to the chiller and is often considered a more complicated task, since its objective stretches well beyond the provision of the desired cooling temperature. Both for reasons of trouble-free operation and energy efficiency, several control loops are often part of the holistic control system, which may influence each other [189, p. 269], making the individual controller

Figure 4.2: Presentation of the test rig and its main components.

design difficult. However, since the objective of the test rig design differs from that of a real-world system, the control strategy is essentially limited to the variables T_{ci} and T_{ei} and results from the considerations presented in [133].

As listed in Table 4.2, the PLC forms the core of the test rig control system by providing multiple communication standards and aggregating all process information. More importantly, it enables the implementation of the control strategy of the chiller system. Likewise, it controls other peripheral devices, such as the position of the three-way valves for controlling the water temperatures as they enter the heat exchangers or the switching of the cooling tower's cooling water pump. Moreover, the system is equipped with a human machine interface (HMI) to be used by the operator to change set-points variables or to monitor the process. The PLC as well as the HMI and the associated electronics are installed within the control cabinet, whereby the overall system is tested and approved by the manufacturer according to the applicable norms and standards.

The test rig illustrated in Figure 4.2 contains a water-glycol mixture being applied as a convector fluid, with ethylene glycol added for corrosion and frost protection as suggested by the system manufacturer. Although the system was initially not expected to yield chilled water-glycol temperatures below 0 °C, it operates close to the freezing point of water at ambient pressure and may drop below it during the start-up sequence or shortly after changing the set-point variables due to time-delays along the closed loop control system. This is especially critical when changing the set-point of T_{ei} (T_{ei}^{*}) from high to low values, which justifies the use of water-glycol rather than water only.

In the final step of the commissioning phase, several tests were performed on the overall system, including software and hardware components, to ensure that the test rig provides its full intended functionality. In a broader sense, this step also includes the determination of the controller parameters for controlling T_{ci} and T_{ei} as well as the operating limits of the chiller, which in turn affect the test sequence to be derived. Thereby, attention is paid to the pressure within the high pressure line p_{rc}, i.e. the refrigerant pressure at

Table 4.2: A collection of the main test rig components.

Subsystem	Component	Qty	Type reference	Manufacturer
Refrigeration cycle	Condenser & Evaporator	2	AlfaNova 76-80H	Alfa Laval
	Oil cooler	1	AlfaNova 52-50H	Alfa Laval
	Screw compressor	1	OSKA5341-K	Bitzer
	Oil separator	1	OA1954A	Bitzer
	Vapour-liquid separator	1	Custom	Pakt GmbH
	Expansion valve	1	HR 2-H for NH3	WITT
Drive technology	Motor	1	5500 LSRPM 160 MP	Leroy Somer
	VFD	1	Powerdrive MDR60T	Leroy Somer
Water-glycol circuits	Pumps	2	Movitec VF 45-1-1	KSB
	Three-way valves	2	Typ 3260/5824	SAMSON
	Cooling tower	1	ATW-24-4G-2	Evapco
	Glycol	1	TYFOCOR	TYFO
Control system	PLC	1	S7-1512C1 PN	Siemens
	HMI	1	Simatic MP 277	Siemens
	IO-Link Master	2	AL1402	ifm

the compressor discharge, since the test rig is equipped with a high pressure emergency shut-down for safety reasons. In fact, this is considered to become critical in case of high cooling water temperatures, high cooling loads or as a result of some induced faults. The reason for this is that these quantities can lead to an increase in the refrigerant condensing temperature T_{rc} and thus also in the pressure p_{rc}. Accordingly, the operating envelope of the chiller is chosen as a compromise between exploiting a wide range of operating conditions and avoiding interruptions within the measurement series triggered by safety devices.

4.2.2 Data Acquisition

Since the goal of this work is to accurately assess faults by analysing the thermodynamic state of a chiller, their representation by the available sensor technology is of great importance to validate the proposed CBM model. Thus, the test rig is equipped with a comprehensive set of sensors introduced for control and state monitoring purposes. Both analogue and digital sensors are included in the system architecture, whereas data are relayed to the PLC via multiple analogue extension modules, a Profibus and a Profinet interface. As shown in Figure 4.3, the latter is used to interface the IO-Link based sensors via the IO-Link Master devices and to aggregate the information provided through the power monitoring system (PMS). To access the information provided by the VFD, it is connected to the PLC through a Profibus fieldbus, which is primarily used for setting and monitoring the compressor speed n_{comp}. Another important communication standard provided is the Open Platform Communication - Unified Architecture (OPC-UA) protocol, as it allows the access of process variables including set-point values, sensor readings, or

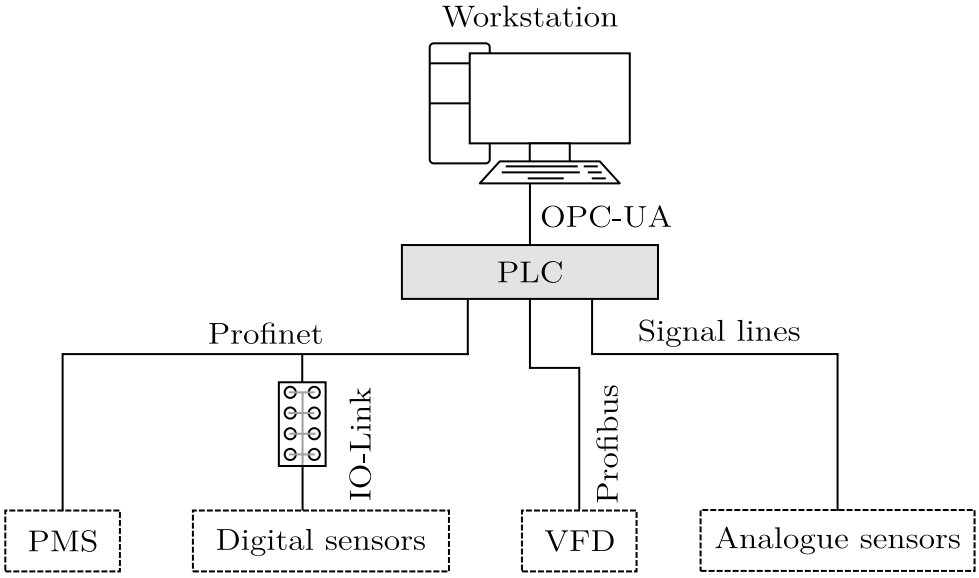

Figure 4.3: Communication architecture for the acquisition of sensor data.

computed quantities. During the test series, the data are acquired from the sensor suite by iteratively requesting data from the PLC integrated OPC-UA server. The counterpart represses a computer deployed for process monitoring purposes, whereby both the cyclic data acquisition and transmission as well as the persistent data storage via a relational database management system are realised by a customised software developed within the scope of this work. Thereby, the data are recorded during the test cases with a sampling interval of 1 sec.

As shown in Table 4.3, the accessible information provided by the test rig is primarily focused on the sensor signals used to monitor its thermodynamic state, most of which are resistance temperature detectors. Following the suggestions of Comstock et al. [133, p. 41], the analogue temperature transmitters were first calibrated with an ice water bath, while the other sensors were already calibrated at the factory and also do not allow any adjustment on site. However, the authors also concluded in their work that for faults assessment purposes absolute accuracy is not as important as precision.

While most temperature sensors are introduced to draw conclusions about the thermodynamic state of the system, those monitoring the water-glycol inlet temperatures at the condenser and evaporator, respectively, are given special attention because they largely determine the operational state of the chiller and therefore serve as process variables to be controlled. As can be seen from the listed sensor suite, the measurements of the evaporator inlet and outlet temperatures on the water-glycol are duplicated for reasons of redundancy, as these values are used to calculate the cooling load, which is often considered a decisive thermodynamic quantity of a chiller system [189, p. 269]. Consequently, some additional redundant sensors are introduced, either for monitoring the stability of the measurements or for safety reasons.

In addition, pressure transducers are included within the sensor suite in both the high pressure line and the suction line within the refrigerant cycle. This is done for monitoring compliance with safety-relevant threshold values during operation. These quantities are furthermore important because they are utilised in this work to generate features from thermodynamic dependencies or from refrigerant property tables. The sensory recording of the volume flows, on the other hand, is used exclusively for the simulation of faults with

Table 4.3: The accessible test rig sensor suite.

Designation	Description	Sensor/Source	Interface	Unit
T_{ci}	Water temperatur at condenser inlet	ETF02 PT100	Analogue	°C
T_{co}	Water temperature at condenser outlet			
T_{ei}	Water temperature and evaporator inlet			
T_{eo}	Water temperature at evaporator outlet			
T_{oil}	Oil feeding temperature			
$T_{amb,i}$	Test rig ambient temperature (inside)			
T_{suc}	Refrigerant suction temperature	AT102	Analogue	°C
T_{dis}	Refrigerant discharge temperature			
T_{ei}^{1}	Water temperature and evaporator inlet	TA2435	IO-Link	°C
T_{eo}^{1}	Water temperature at evaporator outlet			
$T_{amb,o}$	Test rig ambient temperature (outside)			
p_{rc}	Refrigerant pressure at compressor outlet	AKS3000	Analogue	bar
p_{re}	Refrigerant pressure at compressor inlet			
p_{re}^{1}	Refrigerant pressure at compressor inlet	PV7004	IO-Link	bar
$\dot{V}_C$	Cooling water volume flow rate	Proline Promag 50	Analogue	m^3/h
$\dot{V}_E$	Chilled water volume flow rate			
I_{comp}	Instantaneous motor current (compressor)	SENTRON PAC3200 (PMS)	Profinet	A
P_{comp}	Instantaneous motor power (compressor)			kW
n_{comp}	Compressor rotational speed	VFD	Profibus	min^{-1}
$\dot{Q}_C$	Condenser heat flow	PLC[2]		kW
$\dot{Q}_E$	Evaporator heat flow			kW

[1] Redundant sensor used for data validation
[2] Computed variables

predefined severity levels, as described below, and has no direct influence on the CBM model training. Likewise, both energy data via the PMS and the compressor speed via the VFD are acquired and stored, whereby the latter is also used to determine the test sequence, as explained in more detail in the following section.

During operation, the PLC also automatically computed further thermodynamic quantities, namely the condenser and evaporator side heat flow, that cannot directly be measured. Along with the data acquired from the given sensor suite, this information is provided through the PLC, which are used for process monitoring tasks. Thus, the heat flows at the evaporator and the condenser can be computed. For the chilled water-glycol mass flow $\dot{m}_E$ passing through the evaporator, the heat flow absorbed or rejected by that medium is given as [125, p. 127]

$$\dot{Q}_E = \dot{m}_E \Delta h_e = \dot{m}_E c_p \Delta T_e = \dot{V}_E \rho_w c_p (T_{ei} - T_{eo}) \tag{28}$$

where Δh_e is the difference in the specific enthalpy and ΔT_E the difference in water-glycol temperatures before and after passing the evaporator, c_p the specific heat capacity of the respective single-phase medium and ρ_w denotes its density. It goes without saying that this relation similarly holds for both the condenser as well as the evaporator, i.e. $\dot{Q}_C$ and $\dot{Q}_E$ respectively. The heat flow dissipated through the condenser comprises the heat flow absorbed from the chilled system and the compressor power $\dot{Q}_C = \dot{Q}_E + P$. It should

be borne in mind that both c_p and ρ_w slightly differ from those values of water due to the presence of glycol in the water circuits. Nonetheless, for simplicity, the coefficients for water are used in (28) to compute the theoretical heat flow at the condenser and evaporator throughout this work.

In general, all substance-specific coefficients used throughout this dissertation are derived from [190] unless stated otherwise. Although these two features were already computed during the measurement series, most of the computed features used in the following sections have been computed after the data acquisition phase. However, as these quantities play an important role during the test rig commissioning phase as well as for the comparison of the different operating conditions, they are already introduced in this section. For an overall view of the features extracted from the datasets, the reader is referred to Appendix A.5.

4.3 Procedure

This section provides an overview of the procedure for simulating different operational states of the chiller contained within the test rig, whereby the test sequence is firstly derived from the operational envelope. Secondly, steady-state analysis is performed and the applied steady-state criterion is introduced in more detail. Finally, the different classes of the dataset are presented.

4.3.1 Fault Examination

The fault examination procedure described in the following outlines the applied approach to study the effect of some predetermined faults onto the overall chiller operation condition. Consequently, the data acquired during this phase are finally exploited to derive the model in the scope of this dissertation, whereby the overall approach is derived from the one presented by Comstock at al. [133].

As shown in Figure 4.4, during the commissioning of the entire system, all software parts responsible for data storage, control and visualisation as well as all hardware components of the test stand were first tested for faultless functionality. In addition, it had to be ensured that the various operating states could be approached with sufficient repeatability. In the following phase, a benchmark test was performed in order to determine the operating envelope of the system and to determine whether or not the steady-state

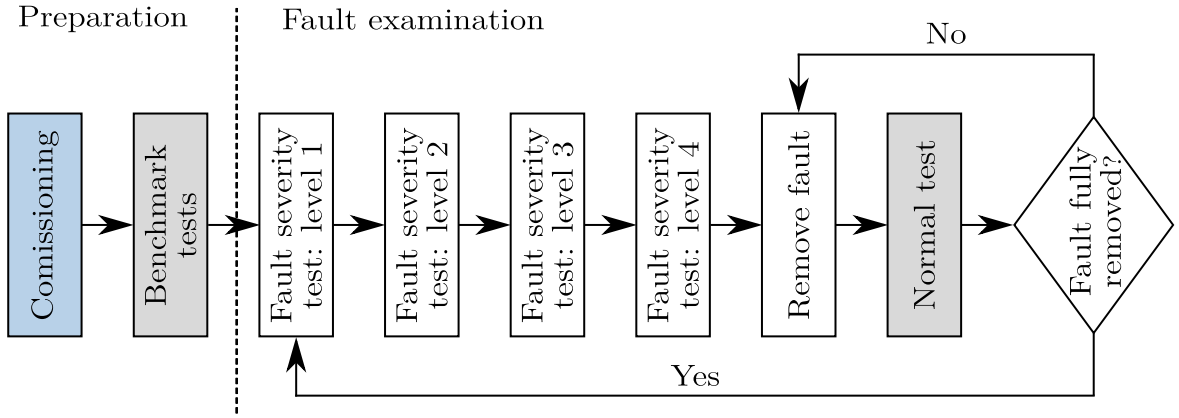

Figure 4.4: The fault examination procedure.

criterion is suitable with regard to the applied control strategy. Most importantly, This test represent the normal operation baseline and is used for recurring data validation along the examination procedure such as, for example, to determine whether the chiller indeed operates under a fault-free condition, i.e. after a previously induced fault has been successfully removed.

During the experiments, only one fault in the system is considered at a time, as faults occurring simultaneously are beyond the scope of this work. To study the effect on the system variables with increasing severity, each fault has been gradually induced by defining four severity levels (SL), approached on after the other. Subsequently, the fault was then removed from the system, which was followed by a normal test run. By comparing the data with the data from the benchmark test, it was then determined whether the fault-free initial state could be restored or traces of the induced fault were still present in the system before starting a new measurement series. If the data of the normal test run showed too large deviations from those of the benchmark test, further measures had to be taken until the fault-free state could be fully restored.

4.3.2 Test Sequence

In order to be able to compare the measurement series, a constant test sequence was defined throughout the experiments, which was reviewed for its feasibility during the commission phase and the benchmark test. This sequence was kept unchanged throughout all measurement series of both faulty and fault-free chiller operations. Based on the concept presented in [133], three control variables were determined making it possible to cover a large operating spectrum of the chiller.

As partly described above, the reference variables T_{ci} and T_{ei} were chosen for this purpose. In addition, the compressor rotational speed n_{comp} is additionally chosen to derive the test sequence shown in Appendix A.2, with the set-points $T^*_{ei} \in \{5, 10, 15\}$, $T^*_{ci} \in \{23, 28, 33\}$ and $n^*_{comp} \in \{65, 50, 35\}$. Consequently, by use of these variables,

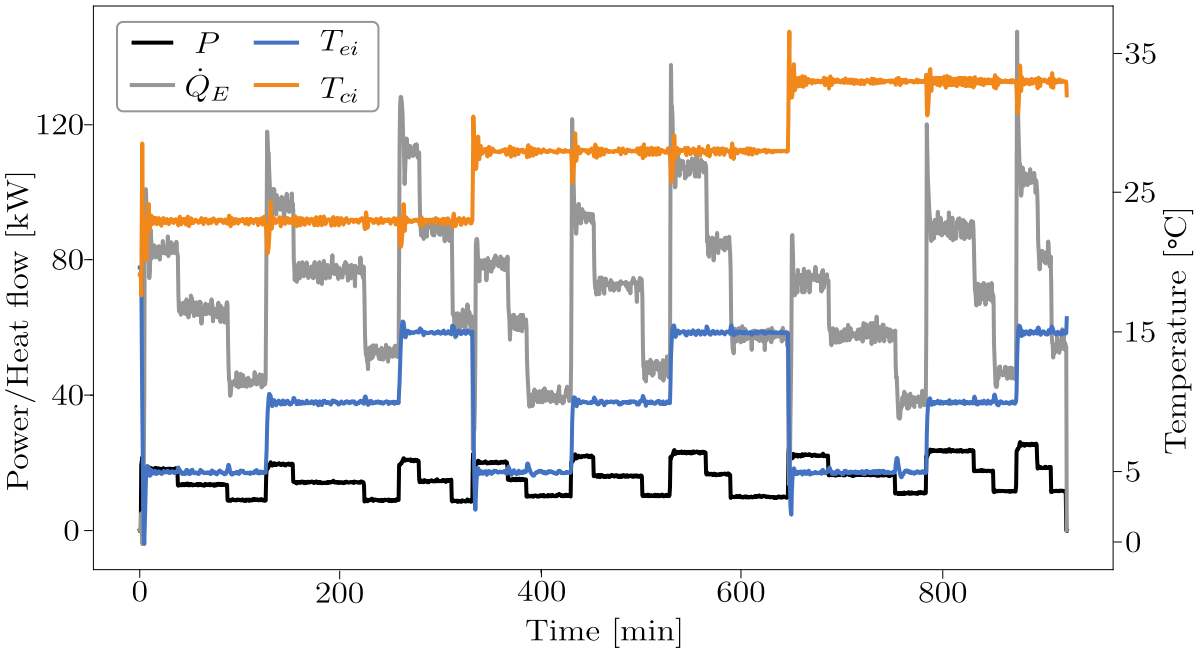

Figure 4.5: The test sequence data plot at normal chiller operation.

a 3x3x3 tensor could be determined representing a total number of 27 system states [133]. As can be seen in Figure 4.5, all predefined operating states were approached one after the other, with each experiment lasting about 12-16 hours before it was completed. Furthermore, a steady-state criterion was introduced that had to be met at least once by each state. Figure 4.5 also depicts the evaporator cooling load $\dot{Q}_E$ as well as the power consumption through the compressor power unit P_{comp} to show the increase in the coefficient of performance ε at low values of T_{ci} and high values of T_{ei}.

4.3.3 Steady-State Detection

The data collected throughout the experiments consists of both transient and a steady-state chiller operating conditions, whereas transient system behaviour is generally considered to occur during start-up or shut-down phases as well as during state transitions, i.e. after changing the set-point variables. As demonstrated in Figure 4.6, such states are often subject to high system dynamics and therefore complicate the fault assessment procedure, as statements about the chiller's thermodynamic state are only possible to a limited extent. Moreover, different dynamic response times may occur across the various chiller subcomponents [191] resulting in time lags at the various sensor locations. As can be seen from the figure, transient data are essentially related to rapidly changing thermodynamic states. Although in this work this is primarily caused by the controllers that control the water-glycol inlet temperatures in order to enable the simulation of various operating states with the test rig presented, transient system states also occur in the real application and must therefore be treated accordingly with regards to CBM. In addition, Figure 4.6 illustrates the issue incurred by the definition of the chiller operating range discussed in the previous sections, as it shows the pressure in the high pressure line p_{rc} following the controlled condenser temperatures while overshooting the targeted value for T_{ci}. This example demonstrates the importance of an appropriate buffer to the operating limits of the chiller when determining the test sequence to avoid triggering the integrated emergency shut-down and thus aborting the experiment.

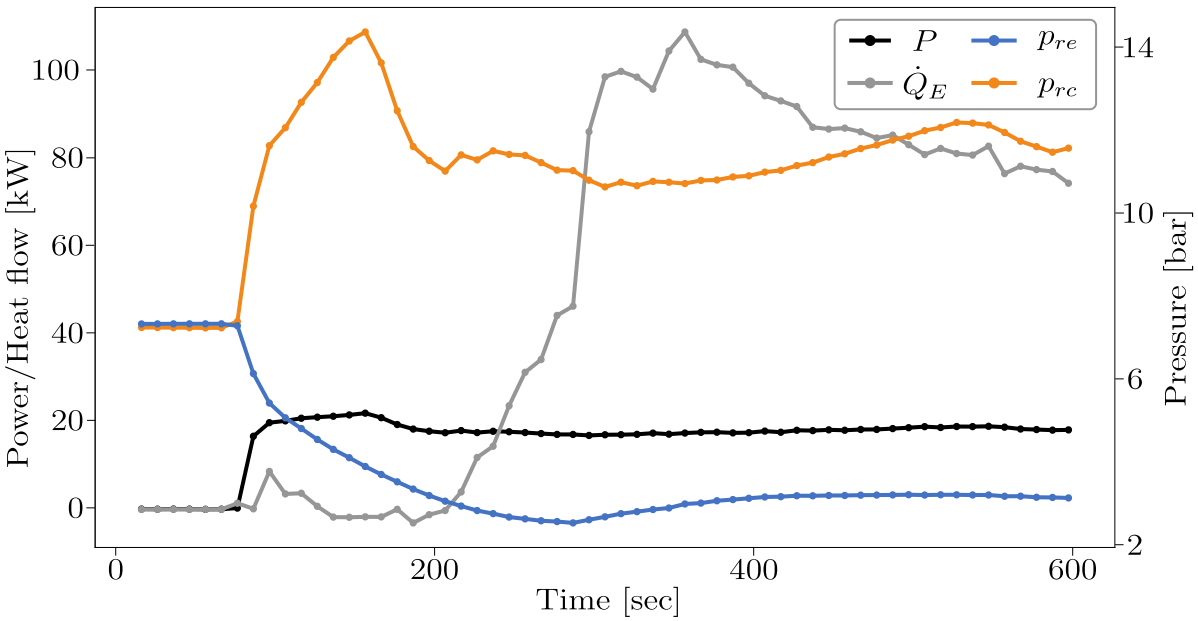

Figure 4.6: Illustration of the transient operating condition during the start-up phase.

Even though some studies exist on the use of transient data for developing a CBM model as described, for example, by Armstrong et al. [192], most authors prefer to follow the harnessing of steady-state data to fulfil this task [6], which is also the focus of this work. However, as no system is ever steady and the respective sensor readings will therefore vary to some degree throughout its operation [193], a meaningful criterion for distinguishing between data stemming from transient system states and from a presumed steady-state must be introduced, for which steady-state filters, or steady-state detectors, are usually employed. As has already been described in Section 3.2, this work follows the approach proposed in [6] for isolating data corresponding to a steady-state. The implemented algorithm within the scope of this work can be found in Appendix A.1.1.

In general, the steady-detector computes local derivatives for each point in time, which are then filtered using a Savitzky-Golay filter, whereby high frequency noise is reduced and the SNR is improved. As this type relies on least-square polynomial approximation within a certain time frame, the polynomial degree pd, the window size ws and an appropriate threshold must be determined. As previously introduced, the latter is defined as some scalar variable δ_{thr} times the standard deviation of the computed local derivatives σ_{dev} following the suggestions of Beghi et al. [6]. As shown in Figure 4.7, a steady

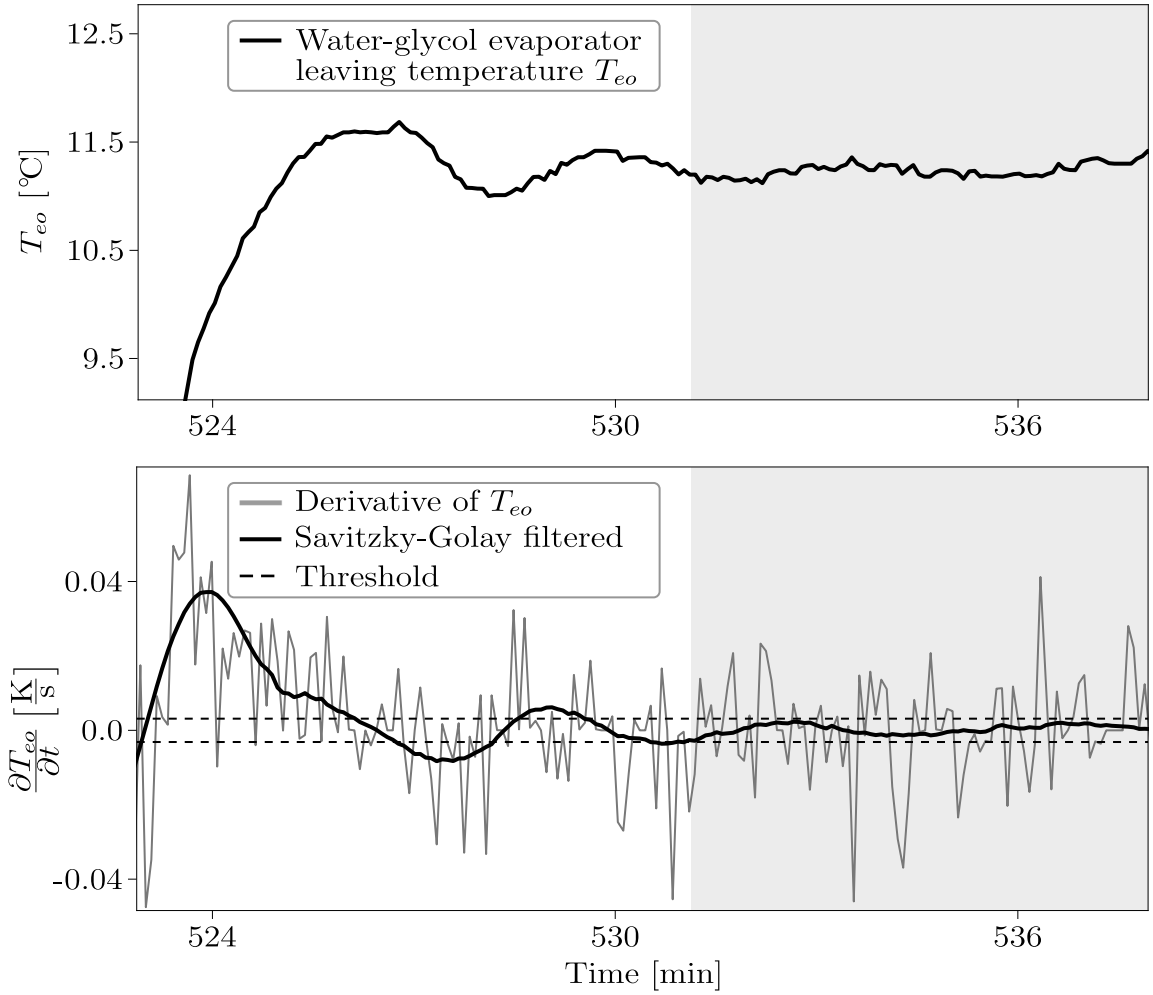

Figure 4.7: Example of chiller approaching steady-state condition (grey area): (upper) time signal of T_{eo} and (lower) steady-state criterion with filtered local derivatives of T_{eo} using the approach of [6].

state is determined when the peaks of the filtered signal are within a predefined threshold for a certain period of time. Furthermore, one must define the variables indicating the steady state. These should be chosen carefully, as a high number of indicators can lead to long estimated transient states and thus reduces the information content exploited within the FDD block [6]. On the other hand, using only one variable may cause inconsistent definitions of the alleged steady-state, since some essential system conditions may not be monitored. As proposed in the original paper, the water-glycol temperature at the evaporator outlet T_{eo} as well as the refrigerant pressure at the lower pressure side of the refrigerant circuit p_{re} are utilised for steady-state indication. Moreover, the following parameters were defined from earlier observations during the test rig commissioning; $pd = 4$, $ws = 241$ and $\delta_t = 0.1$. It should also be noted that a stead-state had to be continuously active for at least 5 minutes ($t_{min} = 5$ min) to be recognised as such.

4.4 Fault Simulation and Patterns

The presented test rig enables the investigation of several faults which, in turn, forms the basis for the data collection phase in this work. Thereby, the faults of interest are artificially induced at different severity levels and monitored within a predefined chiller operating range. Therefore, this section describes the approach towards the fault simulation procedure, whereby the specific fault patterns are presented as determined from the experiments.

4.4.1 Classes

This dissertation focuses on five distinct classes of the collected dataset. The normal, or fault-free, class originates from the test sequences performed during the commissioning phase, the benchmark test and from the successful normal tests after a removing a certain fault. The remaining classes represent faulty operating conditions at different SL, whereas a subset of faults were selected from [133] in this study. This is mainly due to the fact that the proposed CBM model should be applied across multiple domains and, therefore, the same types of faults must be included in the two validation datasets employed in this work.

In addition, the fault types investigated in this work are generally classified as critical, as discussed in Section 2.4.2, and were selected for investigation with the provided test rig on their feasibility. For example, condenser fouling has been identified by Comstock et al. [133] as a severe issue, as it leads to a reduced heat transfer coefficient and thus causes performance losses. The authors simulated this fault type by blocking some water tubes within the shell and tube heat exchanger. Although this does not change the actual heat transfer coefficient, it reduces the heat transfer surface area and therefore induces a similar thermodynamic effect. However, brazed plate heat exchangers were used in the test rig applied in this work, which do not allow the simulation of a reduced heat transfer coefficient in the same manner. This applies equally to refrigerant overcharge simulated in [133], since the test rig used in this work is based on an integrated refrigerant receiver and, thus, a deviation from the normal operational state was only to be expected in the event of liquid refrigerant being drawn by the compressor, an effect also known as liquid slugging. Since screw compressors can only withstand limited amounts of liquid slugging without severe degradation or even destruction [194], this fault was not investigated in this

Table 4.4: Computed thermodynamic variables for state comparison.

Equation	Designation
$T_{re} = \frac{B_A}{A_A - \log_{10}(p_{re})} - C_A$	Refrigerant evaporation temperature
$T_{rc} = \frac{B_A}{A_A - \log_{10}(p_{rc})} - C_A$	Refrigerant condensing temperature
$\Delta T_{ea} = T_{eo} - T_{re}$	Evaporator approach temperature
$\Delta T_{ca} = T_{rc} - T_{co}$	Condenser approach temperature
$\Delta T_E = T_{ei} - T_{eo}$	Evaporator water-glycol temperature difference
$\Delta T_C = T_{co} - T_{ci}$	Condenser water-glycol temperature difference
$\Delta T_{sh,dis} = T_{dis} - T_{rc}$	Superheat temperature at compressor discharge
$\Delta T_{sh,suc} = T_{suc} - T_{re}$	Superheat temperature at compressor suction line
$\varepsilon = \dot{Q}_E / P_{comp}$	Coefficient of performance

work. Furthermore, the simulation of excess oil in the system and a defective expansion valve could not be simulated due to the chiller design characteristics.

As a consequence, four faults were selected to be simulated, namely: (RVE) reduced evaporator water-flow rate, (RVC) reduced condenser water-flow rate, (NC) non-condensable gases and (RL) refrigerant leak. While the first two are component-level faults, as their root causes are locally bounded, the other two are considered system-level faults. Thereby, non-condensable gases usually refer to air being trapped in the refrigerant line, often caused by improperly performed maintenance [133, p. 88]. Refrigerant leak, in turn, refers to the loss of working fluid throughout the chiller operation.

The applied approaches for simulating the faults as well as their impact on the chiller operating condition in comparison to the normal state are presented in more detail below. Besides the thermodynamic state variables presented in Section 4.2.2, the remainder of this section introduces additional features to enable better comparability with the results presented in [133]. It is noteworthy, however, that one can compute them from the information provided by the available sensor suite. As shown in Table 4.4, these variables include, among others, the refrigerant evaporation T_{re} and condensing T_{rc} temperatures, which are approximated using the Antoine equation. It should be noted that the Antoine coefficients A_A, B_A and C_A listed in [195, p. A.59] are used in this work to calculate both the evaporating and condensing temperatures of the circulating refrigerant for the temperature range in which the test rig was expected to be operating. Furthermore, this work applies a multiple linear regression model in a similar manner as in [133] for comparing the various operating conditions as

$$\begin{aligned} y_v(T_{ei}, T_{ci}, \dot{Q}_E) =& \beta_0^* + \beta_1^* \cdot T_{ei} + \beta_2^* \cdot T_{ci} + \beta_3^* \cdot \dot{Q}_E \\ &+ \beta_4^* \cdot T_{ei} \cdot \dot{Q}_E + \beta_5^* \cdot T_{ci} \cdot \dot{Q}_E + \beta_6^* \cdot \dot{Q}_E^2 \end{aligned} \tag{29}$$

whereby y_v can represent any variable from the dataset and β_i^* are the regression coefficients.

As can be seen from Appendix A.2, a total number of 31 experiments were performed, of which 18 were used to extract the dataset. Table 4.5 shows the average deviation δ_e from all normal test cases compared with the benchmark test, whereby the coefficient of determination R^2 is additionally listed to show the quality of the prediction through the

Table 4.5: Average deviations of data of the normal test cases from benchmark dataset.

Variable	BM	N-RVE		N-RVC		N-NC	
	R^2	δ_e	R^2	δ_e	R^2	δ_e	R^2
P_{comp}	0.996	-0.49 %	0.996	1.36 %	0.997	-0.05 %	0.997
p_{re}	0.999	-0.75 %	0.999	-1.18 %	0.998	-0.97 %	0.998
p_{rc}	0.999	-0.30 %	0.999	0.31 %	0.998	0.37 %	0.998
$\Delta T_{sh,suc}$	0.944	-9.50 %	0.935	3.31 %	0.964	12.52 %	0.949
$\Delta T_{sh,dis}$	0.930	-1.10 %	0.850	2.48 %	0.912	12.81 %	0.982
ΔT_{ea}	0.989	3.97 %	0.989	8.37 %	0.989	3.71 %	0.989
ΔT_{ca}	0.996	0.02 %	0.996	0.23 %	0.997	-1.35 %	0.997
ΔT_E	0.999	0.37 %	0.999	1.22 %	0.999	1.34 %	0.999
ΔT_C	0.989	-1.77 %	0.990	1.78 %	0.995	5.23 %	0.996
ε	0.947	0.58 %	0.946	0.52 %	0.958	2.26 %	0.959
T_{oil}	0.995	-0.89 %	0.992	0.81 %	0.994	2.52 %	0.991

regression model (29). One may expect from this table a slight impact of seasonal trends on the entire measurement series, as the data collection phase took about 7 months starting in winter and ending in autumn. For example, the benchmark test 'BM' was conducted about 5 months earlier than the normal test case 'N-NC', which led to slight deviations of some sensor values between the two experiments due to the varying ambient temperatures that affected the measurement series. In this regard, it should be noted that the deviations of $\Delta T_{sh,suc}$ and $\Delta T_{sh,dis}$, although appearing high, only amount to a difference of less than ± 0.4 K, with the only exception of $\Delta T_{sh,dis}$ in the 'N-NC' dataset, which is about $+1.67$ K. Nevertheless, the repeatability appears to be stable under the circumstance of varying ambient temperatures of $T_{amb,i}$ ranging from 18 °C to 34 °C and $T_{amb,o}$ ranging from 0 °C to 35 °C throughout the conducted experiments. As will be pointed out in the following, the impact of the investigated faults on these state variables is significantly larger compared to the experimental uncertainty.

The following sections provide a detailed overview of the faults induced in this study, the definition of their respective severity levels and their impact on the overall operating condition. Besides the experimental results depicted subsequently, the reader is also referred to Appendix A.4 for further details. It should be further noted that the evaluation points of respective regression models shown in the figures below, are all evaluated at $T_{ei}^* = 10$ °C for simplicity reasons. The results can be directly compared to the findings presented in [133] and show that both Hypothesis H1.1 (varying fault characteristics across domains), but also Hypothesis H3.2 (existence transferable fault indicative features) can be confirmed.

4.4.2 Reduced Evaporator Water-Flow Rate

In this work, the fault RVE was induced by reducing the water-flow rate through the evaporator $\dot{V}_E$ through the integrated ball valve in the chilled water circuit (see Figure 4.1). As both integrated pumps are pressure controlled, the flow rate could be decreased by carefully adjusting the valve position which, due to the change in pressure differences, led to a reduction in the flow rate. Furthermore, the flow rate could be tuned by directly utilising the respective sensor readings from the electromagnetic flowmeters integrated in

Table 4.6: The severity gradation of the reduced evaporator flow rate.

Test	Test scenario	Desired condition	Actual condition
0	Severity Level 1	15 % reduced flow rate	20.8 m^3/h
1	Severity Level 2	30 % reduced flow rate	17.0 m^3/h
2	Severity Level 3	45 % reduced flow rate	13.6 m^3/h
3	Severity Level 4	60 % reduced flow rate	9.9 m^3/h
4	Normal		25.0 m^3/h

Table 4.7: The average deviations of RVE datasets from benchmark dataset.

Variable	RVE1		RVE2		RVE3		RVE4	
	δ_e	R^2	δ_e	R^2	δ_e	R^2	δ_e	R^2
P_{comp}	-1.22 %	0.997	0.09 %	0.996	3.39 %	0.997	6.59 %	0.997
p_{re}	-1.34 %	0.999	-3.08 %	0.999	-5.75 %	0.998	-9.63 %	0.998
p_{rc}	-0.13 %	0.998	-0.10 %	0.999	-0.01 %	0.998	0.29 %	0.999
$\Delta T_{sh,suc}$	-3.93 %	0.915	-2.80 %	0.909	-4.07 %	0.937	2.51 %	0.936
$\Delta T_{sh,dis}$	1.08 %	0.890	-5.61 %	0.830	-2.33 %	0.687	0.89 %	0.826
ΔT_{ea}	-2.10 %	0.989	-4.46 %	0.988	-8.78 %	0.990	-16.96 %	0.987
ΔT_{ca}	-0.30 %	0.997	0.27 %	0.997	0.74 %	0.998	1.41 %	0.999
ΔT_E	18.53 %	0.999	42.87 %	0.999	79.54 %	0.999	147.16 %	0.999
ΔT_C	-1.32 %	0.994	-1.27 %	0.993	-1.35 %	0.996	-0.90 %	0.997
ε	2.01 %	0.946	0.76 %	0.961	-2.77 %	0.952	-4.23 %	0.952
T_{oil}	-0.54 %	0.994	-0.49 %	0.987	-0.39 %	0.992	0.53 %	0.997

the chilled water circuit. As shown in Table 4.6 the flow rate was gradually decreased in 15 % steps starting from the initial rate of $\dot{V}_E = 25.0$ m^3/h.

As can be seen from Table 4.7 as well as Appendix A.4.1, this fault is i.a. characterised by rising values of ΔT_E, as can be directly inferred from (28). Accordingly, the temperature drop causing the trend of p_{re} was expected. Figure 4.8 illustrates the effect of increasing SL on the evaporator pressure, which decreased analogously to the evaporating temperature T_{re}, especially at high cooling loads. It was further observed that this fault is accompanied by an increased power consumption P_{comp} as it resulted in lower evaporator outlet temperatures T_{eo} which in turn affected the refrigerant evaporating temperature T_{re}. Consequently, the chiller had to maintain a lower pressure within the evaporator, whereby the pronounced pressure difference between the suction and high pressure line led to an increase in the overall power consumption. However, one should bear in mind that these values are slightly larger than the measurement uncertainty and should therefore be interpreted with caution.

Although these patterns are similar to those presented in [133], they still differ in their characteristics. In particular, the difference in the evaporator approach temperatures ΔT_{ea} should be underlined in this respect. As can be seen from Figure 4.9, ΔT_{ea} decreases at higher SL, whereby this characteristic is particularly pronounced at lower chiller cooling loads. Interestingly, this contrasts with results presented by Comstock et al. [133], where ΔT_{ea} is increased while reducing the evaporator water flow rate. This seems to be mainly due to the use of different types of heat exchangers within the two test rigs. The operation of the shell and tube heat exchanger utilised within the ASHRAE 1043-RP project is

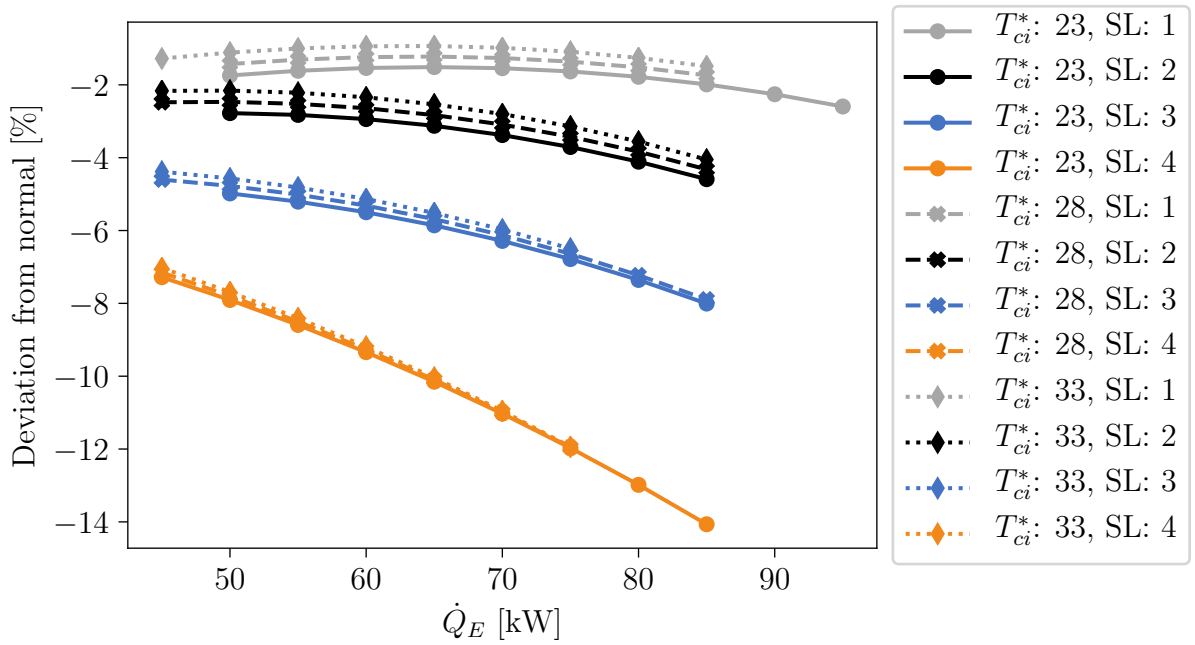

Figure 4.8: The deviation of p_{re} for RVE.

differently affected by this fault compared to the plate heat exchanger, which exemplary confirms Hypothesis H1.1. As the latter provides a highly turbulent flow, a smaller hydraulic diameter and a comparatively higher effective heat transfer area, it promotes an enhanced heat transfer coefficient [196, pp. 6-8], even at lower water flow rates. Most importantly, the chiller used in the test rig integrates a flooded evaporator, allowing lower evaporator approach temperatures to be achieved [197, pp. 476-478] compared to its dry expansion counterpart used in [133]. Consequently, the lower flow velocity caused by the lower flow rate allowed more time for heat exchange, resulting in lower values of ΔT_{ea}, whereas by use of a shell and tube type evaporator this value is likely to be increased

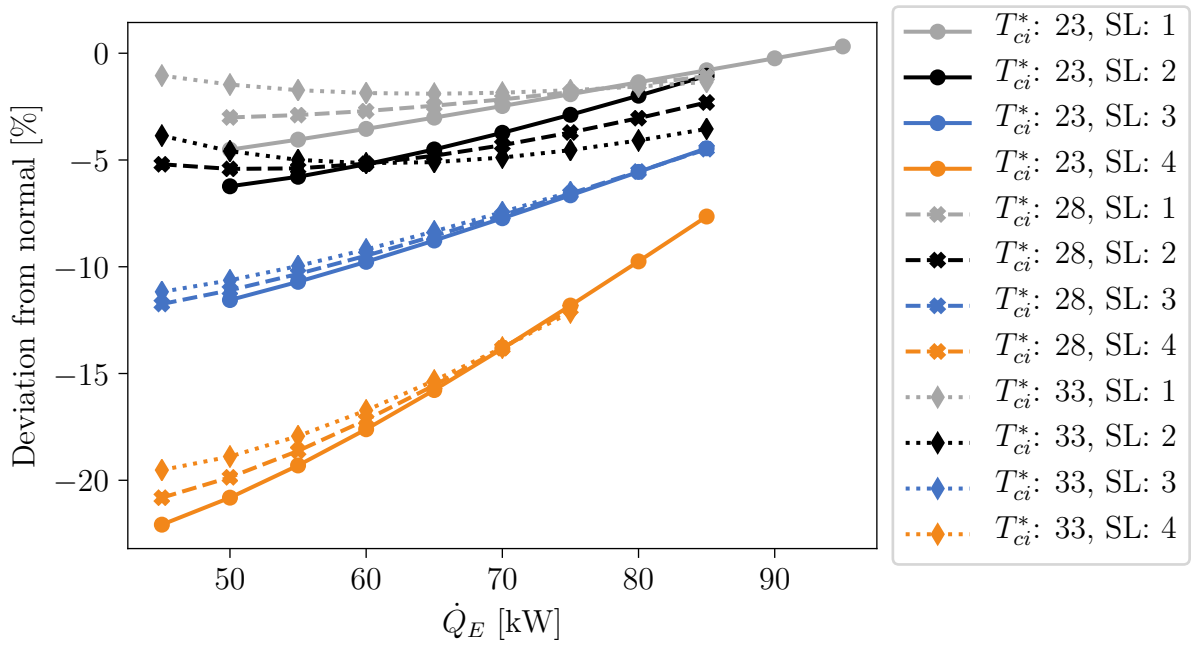

Figure 4.9: The deviation of ΔT_{ea} for RVE.

due to the degraded heat transfer between the tube wall surface and the working fluid. This once again underlines a key assumption made at the beginning of this dissertation, i.e. that the transferability of certain fault characteristics amongst heterogeneous chiller systems may not be possible to a full extent.

4.4.3 Reduced Condenser Water-Flow Rate

Similar to the previous considerations regarding RVE, this fault was induced by reducing the water-flow rate through the condenser, which was tuned by gradually closing the respective ball-valve while monitoring the measured volume flow rate provided through the flow meter within the cooling water circuit. Compared to the previous fault, a high-pressure emergency shut-down was likely during the measurement series when investigating this fault, especially at high water inlet temperatures at both the evaporator and the condenser. This is mainly due to the fact that higher pressure values within the high pressure line of the investigated chiller were to be expected with this fault as a direct result of the increased refrigerant condensing temperatures T_{rc}. Thus, a smaller SL fault gradation had to be chosen, as shown in Table 4.8, to avoid interruptions during the data collection. Besides, the reduction in water flow through the condenser resulted in increased power consumption by the chiller, which is shown in Table 4.9.

A further similarity can be identified in the risen difference between the condenser inlet and outlet temperatures of the cooling water ΔT_C. As illustrated in Figure 4.10, the deviation compared to the benchmark dataset was more than 35 % for the highest

Table 4.8: The severity gradation of the reduced condenser water-flow rate.

Test	Test scenario	Desired condition	Actual condition
0	Severity Level 1	7.5 % reduced flow	20.5 m^3/h
1	Severity Level 2	15.0 % reduced flow	18.6 m^3/h
2	Severity Level 3	22.5 % reduced flow	17.0 m^3/h
3	Severity Level 4	30.0 % reduced flow	15.4 m^3/h
4	Normal		22.0 m^3/h

Table 4.9: The average deviations of RVC datasets from the benchmark dataset.

Variable	RVC1		RVC2		RVC3		RVC4	
	δ_e	R^2	δ_e	R^2	δ_e	R^2	δ_e	R^2
P_{comp}	-0.17 %	0.996	1.25 %	0.996	5.20 %	0.995	5.88 %	0.994
p_{re}	-0.62 %	0.998	-0.55 %	0.999	-0.70 %	0.999	-0.81 %	0.998
p_{rc}	0.53 %	0.999	1.90 %	0.998	3.51 %	0.998	4.91 %	0.998
$\Delta T_{sh,suc}$	-5.19 %	0.879	-7.37 %	0.938	-6.10 %	0.959	-6.83 %	0.912
$\Delta T_{sh,dis}$	-4.77 %	0.775	-5.09 %	0.764	-7.00 %	0.868	-3.89 %	0.665
ΔT_{ea}	2.26 %	0.988	3.26 %	0.988	5.33 %	0.987	4.55 %	0.986
ΔT_{ca}	0.93 %	0.996	2.20 %	0.996	3.19 %	0.995	3.55 %	0.995
ΔT_E	0.11 %	0.999	0.19 %	0.999	0.34 %	0.999	0.19 %	0.999
ΔT_C	3.29 %	0.990	12.76 %	0.988	24.92 %	0.987	35.55 %	0.988
ε	1.21 %	0.959	0.09 %	0.940	-5.33 %	0.949	-4.44 %	0.958
T_{oil}	-0.54 %	0.986	0.44 %	0.988	1.73 %	0.987	2.76 %	0.984

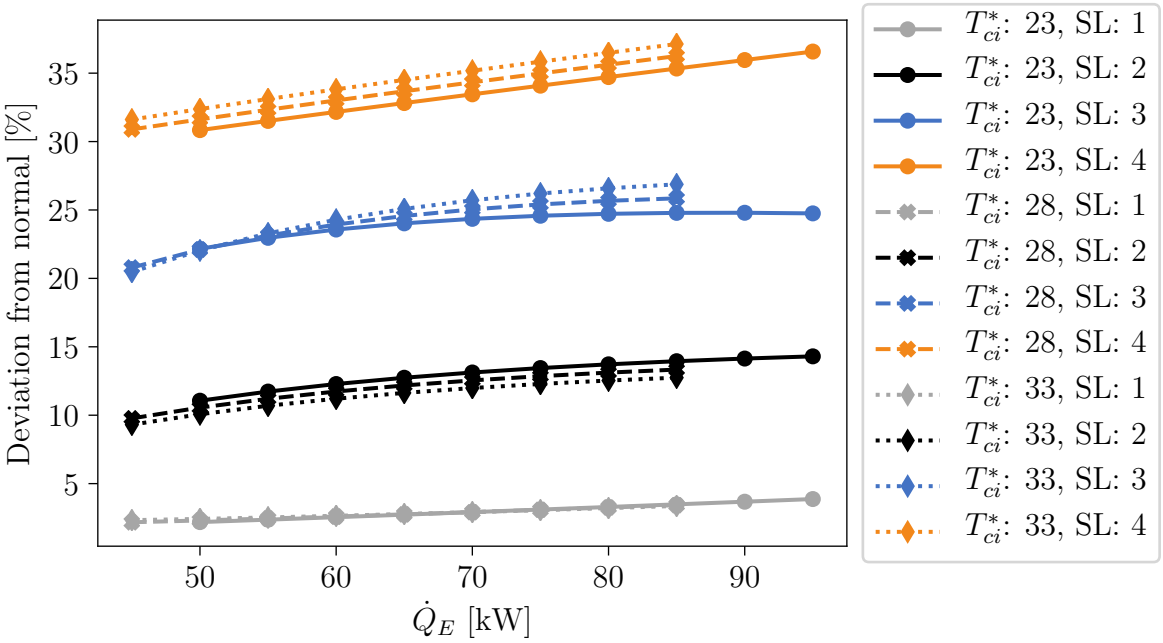

Figure 4.10: The deviation of ΔT_C for RVC.

SL. As a consequence, the refrigerant condensing temperature T_{rc} was similarly increased due to the change in the water condenser outlet temperature T_{co}. From Figure 4.11, one can additionally see the impact on the pressure within the high pressure line p_{rc}, which as expected was increased especially at high chiller cooling loads. Furthermore, Appendix A.4.2 shows slightly higher oil temperatures T_{oil} as well as an increased condenser approach temperature ΔT_{ca}, whereby the higher measurement uncertainty should be taken into account when interpreting the latter. In addition, the reduction of the water-flow through the condenser caused an increased chiller power consumption, which is depicted in Table 4.9. A further similarity can be identified in the risen difference

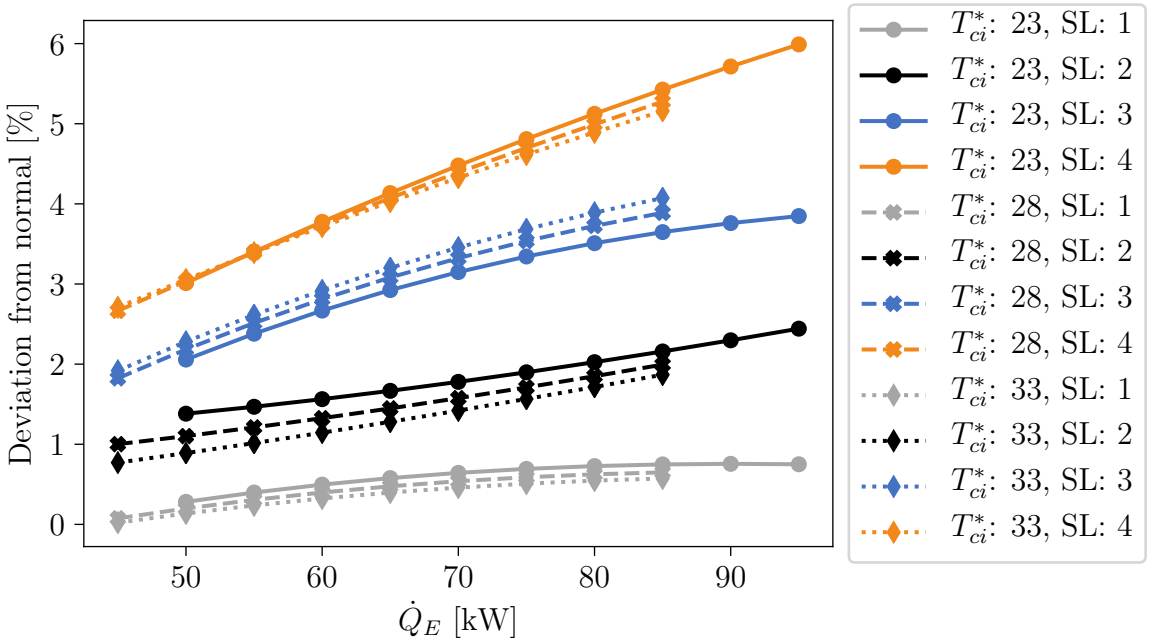

Figure 4.11: The deviation of p_{rc} for RVC.

between the condenser inlet and outlet temperatures of the cooling water ΔT_C. As illustrated in Figure 4.10, the deviation compared to the benchmark dataset was more than 35 % for the highest SL. As a consequence, the refrigerant condensing temperature T_{rc} was similarly increased due to the change in the water condenser outlet temperature T_{co}. From Figure 4.11, one can additionally identify that impact on the pressure within the high pressure line p_{rc}, which as expected was increased especially at high chiller cooling loads. Furthermore, Appendix A.4.2 shows slightly higher oil temperatures T_{oil} as well as an increased condenser approach temperature ΔT_{ca}, whereby the higher measurement uncertainty should be taken into account when interpreting the latter.

4.4.4 Non-Condensable Gases

The fault NC is commonly caused by entrapped air within the refrigerant line. To simulate this condition, predefined amounts of nitrogen were injected into the high pressure line. Since its chemical properties are similar to those of air, comparable results can be achieved without the risk of moisture contamination [133, p. 88]. Table 4.10 shows the selected severity gradation by the gradual addition of nitrogen. As the nitrogen was expected to be mainly trapped in the condenser, the chosen SL were based on the theoretical displacement of the refrigerant by the induced nitrogen within the heat exchanger. The calculation of the displaced volume was based on the mean pressure p_{rc} and the corresponding refrigerant condensing temperature T_{re} prevailing in the normal operating condition of the chiller using the data from the benchmark test. Accordingly, SL1 corresponds to approx. 10 % of the condenser volume at the computed mean pressure, SL2 to approx. 20 % of the condenser volume, etc.

However, as was recognised throughout the experiments, the nitrogen was not only located within the condenser, but was also trapped within the expansion valve, resulting in anomalous system behaviour even after the high-pressure line was completely evacuated and refilled with refrigerant. Only after the explicit evacuation of the expansion valve, the initial chiller operating state could be restored. In order to determine the amount of gas injected into the system, the weight of the nitrogen gas cylinder was continuously monitored by use of a precision scale with a measurement resolution of ±0.1 g. Yet, this also led to inconsistencies in the reproducibility, as the exact determination of the cylinder weight during the filling process was only possible to a limited extent, which explains the sometimes larger deviations between the SL increments. In general, this fault caused the most severe impact on the operating conditions of the overall system in all the experiments carried out. Thus, the fault patterns it caused are comparatively easy to be identified, as can be seen in Table 4.11. Thus, SL1 already led to a significant increase in pressure within the high-pressure line, which is why the operating condition in Test 25 (see Appendix A.2)

Table 4.10: The severity gradation of non-condensable gases in refrigerant line.

Test	Test Scenario	Desired Condition	Actual condition
0	Severity Level 1	Add 10 % by volume nitrogen	17.2 g added
1	Severity Level 2	Add 20 % by volume nitrogen	38.2 g added
2	Severity Level 3	Add 30 % by volume nitrogen	53.5 g added
3	Severity Level 4	Add 40 % by volume nitrogen	68.2 g added
4	Normal	No nitrogen present	

Table 4.11: The average deviations of NC datasets from the benchmark dataset.

Variable	NC1		NC2		NC3		NC4	
	δ_e	R^2	δ_e	R^2	δ_e	R^2	δ_e	R^2
P_{comp}	7.46 %	0.995	18.02 %	0.995	20.61 %	0.995	27.69 %	0.993
p_{re}	-1.09 %	0.998	-2.06 %	0.999	-1.23 %	0.999	-1.33 %	0.999
p_{rc}	7.10 %	0.998	16.42 %	0.996	21.23 %	0.993	26.97 %	0.995
$\Delta T_{sh,suc}$	11.54 %	0.960	8.75 %	0.927	-9.15 %	0.879	-15.24 %	0.950
$\Delta T_{sh,dis}$	-7.17 %	0.944	-6.36 %	0.825	-13.62 %	0.762	-35.22 %	0.717
ΔT_{ea}	7.15 %	0.989	14.98 %	0.992	11.74 %	0.994	10.58 %	0.988
ΔT_{ca}	41.69 %	0.996	97.45 %	0.994	127.19 %	0.995	166.18 %	0.995
ΔT_E	1.40 %	0.999	1.28 %	0.999	0.73 %	0.999	0.69 %	0.999
ΔT_C	4.01 %	0.993	5.44 %	0.995	2.55 %	0.996	1.97 %	0.993
ε	-5.02 %	0.955	-16.31 %	0.958	-20.09 %	0.973	-26.93 %	0.967
T_{oil}	5.32 %	0.993	8.01 %	0.983	6.71 %	0.981	7.58 %	0.975

could no longer be approached due to the risk of an emergency shut-down. As illustrated in Figure 4.12, the pressure continuously increased with higher SL leading to fewer and fewer reachable operating states, especially at high condenser water inlet temperatures. The test scenario to investigate the highest SL consequently only allowed 17 operating states to be approached, whereby high cooling loads in particular had to be avoided. As expected, the differences in the high pressure range are mainly due to the nitrogen located within the condenser. Since the gas occupies a certain volume inside the heat exchanger, the heat transfer area available for refrigerant condensation is reduced. Since the expansion valve controls the refrigerant to be completely liquefied after passing the condenser, the corresponding highside float regulator causes an increase in the pressure ratios.

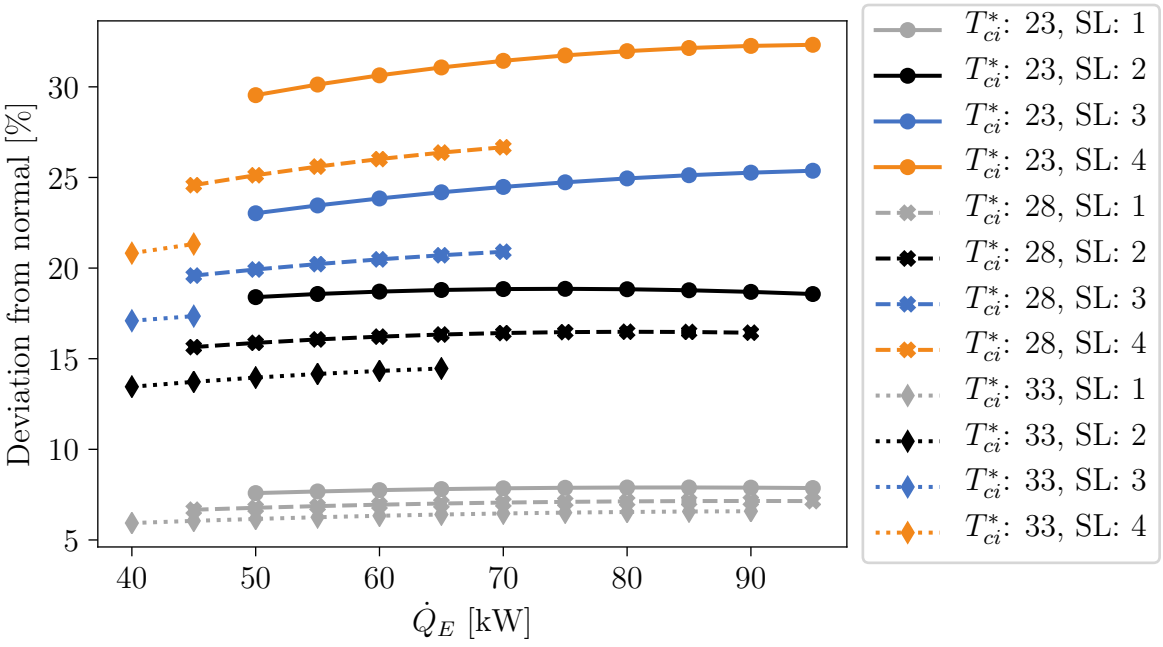

Figure 4.12: The deviation of p_{rc} for NC.

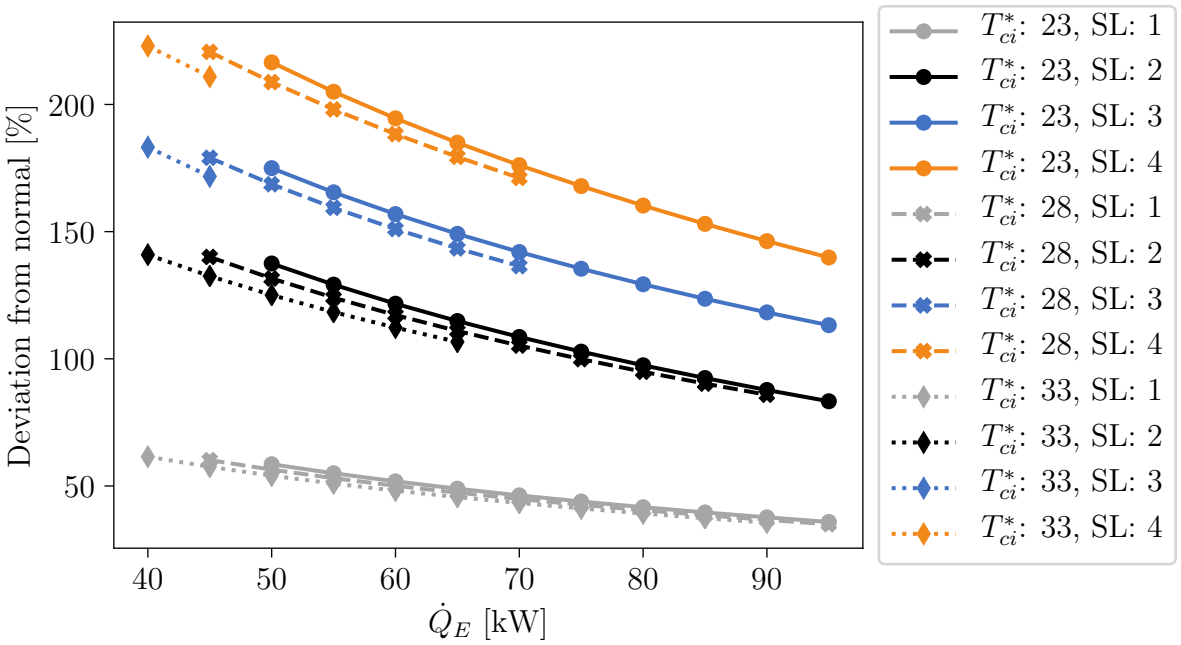

Figure 4.13: The deviation of ΔT_{ca} for NC.

Furthermore, this fault appears to be associated with a drop in the refrigerant superheat temperature at the compressor discharge $\Delta T_{sh,dis}$. Although the regression model does not seem to approximate the underlying data structure of this variable well, as shown by the lower values of R^2, a clear trend is evident. Appendix A.4.3 shows that $\Delta T_{sh,dis}$ was only slightly affected in SL1, but decreased as more nitrogen was induced into the system, with deviations from the benchmark dataset of more than 35 % observed in SL4. Although this property seems unambiguous for this type of fault, it is somewhat misleading from a thermodynamic point of view. Thus, the reason for the drop in $\Delta T_{sh,dis}$ is essentially the increasing heat flow over the oil cooler as a result of the control mechanism designed to stabilise the oil feeding temperature T_{oil}. Due to the increased pressure lift between the suction and high-pressure lines, the compressor is subject to higher power consumption and, therefore, the coefficient of performance ε is significantly reduced. One can furthermore recognise a significant difference in the condenser approach temperature ΔT_{ca}, as shown in Figure 4.13. From the regression analysis presented in this section, it appears that the fault patterns are the most pronounced for lower condenser water inlet temperatures T_{ci}. Moreover, it can be concluded that, although some patterns show similar characteristics to the fault RVC, the deviations from the benchmark dataset of most features appear to be larger at low chiller cooling loads.

4.4.5 Refrigerant Leak

The last fault investigation was carried out during the test rig dismantling phase and could therefore be realised with little effort. Due to the behaviour of the flooded evaporator in case of refrigerant loss, a significant change in the operating conditions was only to be expected when the refrigerant level fell below a certain limit. Since this limit was not known, instead of defining a set of predefined SL, the amount of refrigerant was gradually reduced until a noticeable change became obvious, as shown in Table 4.12. Consequently, a decisive fault pattern could be recognised after 50 % of the working fluid

Table 4.12: The severity gradation of refrigerant leak.

Test	Test Scenario	Desired Condition	Actual condition
0	Severity Level 0	10 % reduced refrigerant	12.1 kg
1	Severity Level 0	20 % reduced refrigerant	10.6 kg
2	Severity Level 0	30 % reduced refrigerant	9.2 kg
3	Severity Level 0	40 % reduced refrigerant	7.8 kg
4	Severity Level 1	50 % reduced refrigerant	6.4 kg
5	Severity Level 2	60 % reduced refrigerant	5.0 kg

had been removed from the chiller. On the other hand, the removal of 60 % caused the entire system to fail and only the operational states at low compressor rotational speed n_{comp}, n^*_{comp} could be approached. Thus, only two SL can be identified in this fault class, as shown in Table 4.13. For the deviations of all RL datasets, the reader is referred to Appendix A.4.5.

It should be noted that this fault was investigated after the system was fully evacuated and refilled again to determine the quantity of working fluid left within the refrigerant line, as removing the previous fault also caused some refrigerant loss. As a consequence, this led to slight deviations between the measured values compared to the benchmark test. Furthermore, these tests were carried out in midsummer, approx. 4 months after the last fault test and approx. 6 months after the benchmark test with outside temperatures of up to 35 °C. As a result, a comparatively larger deviation could be recognised throughput these experiments, as shown in the figures below. Even though the regression model was able to approximate the collected data well, as becomes evident from the high values of R^2, SL2 could only be investigated at rather low cooling loads and therefore a meaningful interpretation of a characteristic trend would barely be possible. Accordingly, the illustration of the associated observations shows only the mean values of the specific operating conditions. As illustrated in Figure 4.14, a significant characteristic of this fault is the increasing evaporator approach temperature ΔT_{ea}. On the other hand, ΔT_{ca} does not seem to be affected and therefore represents a discrepancy compared to the centrifugal

Table 4.13: The average deviations of RL datasets from the benchmark dataset.

Variable	RL1		RL2	
	δ_e	R^2	δ_e	R^2
P_{comp}	19.08 %	0.988	49.82 %	0.990
p_{re}	-19.28 %	0.965	-44.32 %	0.990
p_{rc}	0.79 %	0.998	0.80 %	0.999
$\Delta T_{sh,suc}$	238.48 %	0.905	468.97 %	0.910
$\Delta T_{sh,dis}$	21.67 %	0.930	78.52 %	0.934
ΔT_{ea}	323.52 %	0.954	1422.07 %	0.997
ΔT_{ca}	1.73 %	0.994	0.69 %	0.930
ΔT_E	0.12 %	0.999	-0.38 %	0.999
ΔT_C	4.81 %	0.991	14.11 %	0.880
ε	-14.70 %	0.969	-33.24 %	0.990
T_{oil}	6.44 %	0.993	12.95 %	0.998

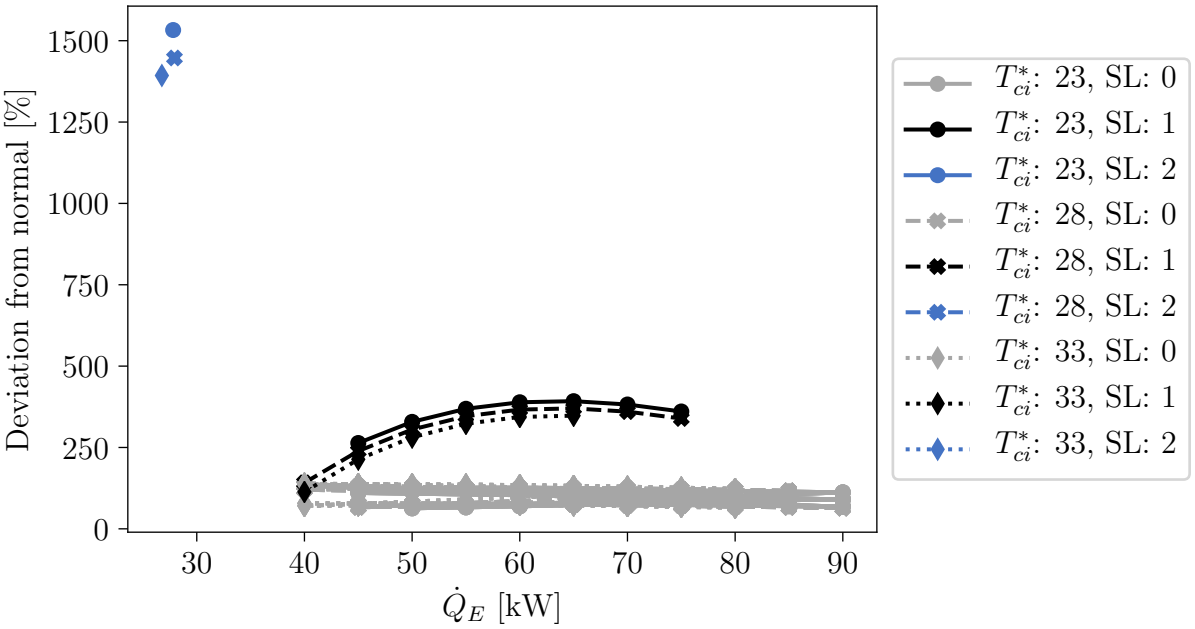

Figure 4.14: The deviation of ΔT_{ea} for RL.

chiller dataset. Another fault sensitive feature has turned out to be the refrigerant suction superheat temperature $\Delta T_{sh,suc}$, as shown in Figure 4.15. This is an important indication especially because refrigerant superheat on the low pressure side is likely to be minimal or non-existent in a flooded evaporator design. As highlighted in Appendix A.4.3, the fault is accompanied by a pressure drop of p_{re} and further results in a lower coefficient of performance ε as indicated due to the increased power consumption caused by the severely increased pressure drop. Lastly, the superheat temperature at the compressor discharge $\Delta T_{sh,dis}$ is slightly higher compared to the normal operating condition. Yet, this should be interpreted with caution due to the implemented oil temperature control loop.

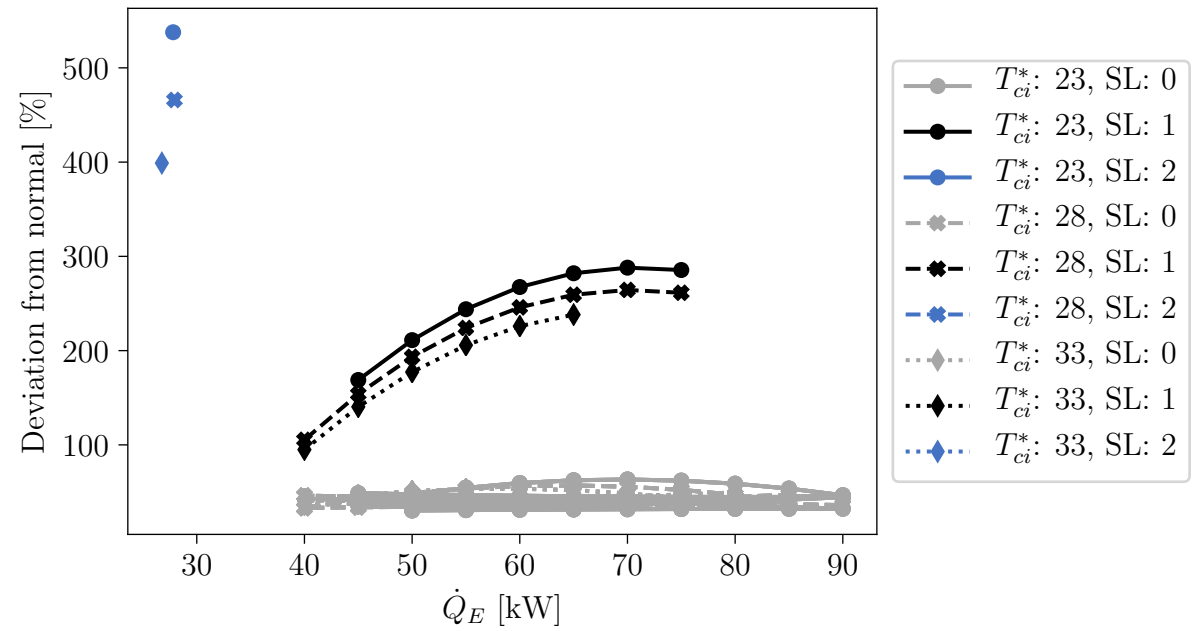

Figure 4.15: The deviation of $\Delta T_{sh,suc}$ for RL.

5 Model Validation

The evaluation of the CBM model performance is of great importance to determine its general applicability. In order to reveal its strengths but also its weaknesses, it is necessary to conduct a comprehensive validation using certain performance measures, the task of which is the essential content of this chapter. To this end, the evaluation concept is presented first, followed by the introduction of the preconditions, such as the chosen feature space as well as the applied metrics. Subsequently, the individual FDD results of the fault detection, the fault isolation and the fault identification layer are reviewed and evaluated accordingly. Finally, the performance of the holistic model is assessed and the results are presented. Throughout this chapter, the research questions presented in Section 2.6 as well as their accompanying hypotheses will be specifically dealt with.

5.1 Concept

When considering testing procedures for software or software parts, a general distinction is made between the terms *verification* and *validation*, whereas the former generally refers to the examination of whether the respective product complies with the given specification. However, verifying the individual facets of ML models may only be feasible to a limited extent because humans may not be able to explicitly describe the expected system behaviour [198]. Validation, in contrast, is intended to show whether the system specification meets the customer requirements or, in the context of data-driven methods, whether the model has learned the desired behaviour. According to Hand and Khan [199], this term is more commonly used in the context of ML to ensure that a model's classification results are sufficiently accurate for the specific problem and, hence, whether it learned the discriminative structure from the data. This term will therefore be used throughout this chapter.

As shown in Figure 5.1, the model validation phase is divided into two consecutive steps. First, each layer within the FDD block is reviewed, with the respective parameter optimisation tasks being performed independently. This also implies that the first layer is validated for its ability to detect novelties by use of target domain test data. Thereby, the actual labels related to the fault classes play only a subordinate role, as this layer deals with a binary decision problem only (deciding between a faulty or fault-free condition). Nonetheless, for reasons of generality, the fault detection performance is also evaluated separately for each fault case, i.e. the algorithm is validated for each fault in the dataset individually. The validation phase of the second layer, on the other hand, is performed to asses the fault classification correctness, whereby only fault-associated data are considered for testing. Furthermore, the analysis of the fault identification layer shall confirm the assumption of a dependency between the presented statistical distance metric and the fault severity. Referring to Section 3.6.1, one can see the relation between the first and second FDD layer, namely the effect of the fault detection performance on the functionality of the fault isolation algorithm. As a consequence, the classification performance of the holistic model will be considered in the closing section.

Another important aspect of the validation phase concerns the test dataset, which, although originating from the same distribution as the target domain training data, is extracted separately. As will be pointed out in Section 5.2.3, splitting the available data into a training dataset and test dataset serves to provide a general indication of the model's generalisation error [200, pp. 119]. Thereby, the term generalisation represents

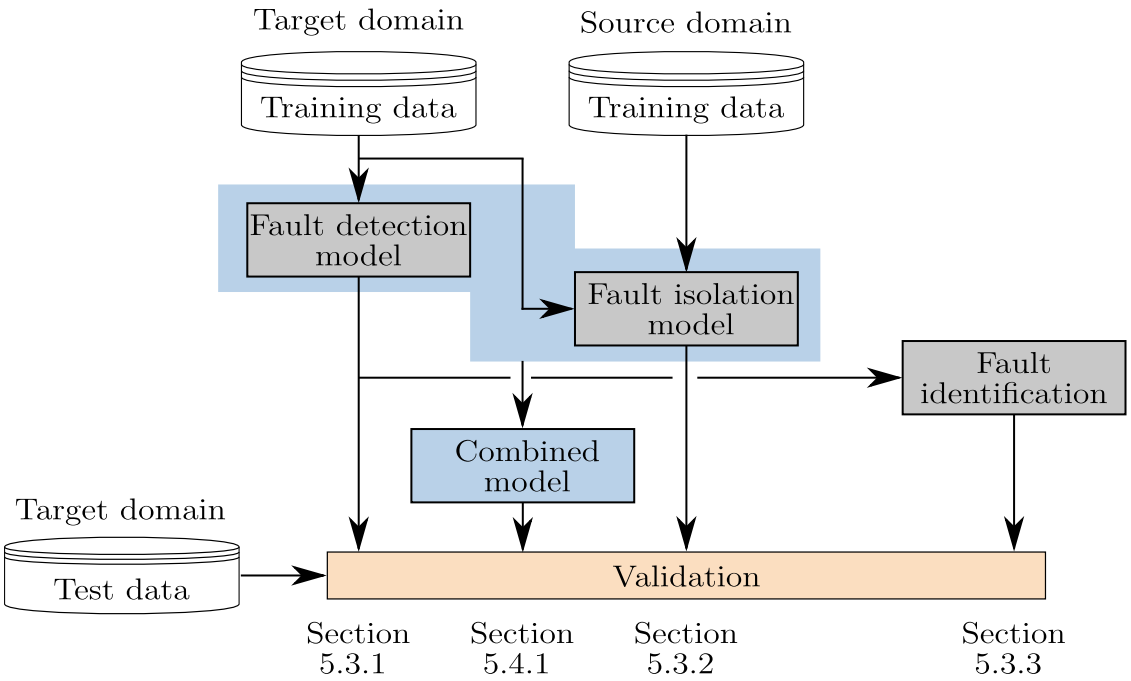

Figure 5.1: Overview of the validation approach and the associated sections.

also an important keyword with regard to the heterogeneity of the various chiller systems, especially concerning Hypothesis H1.1 and Hypothesis H1.2, and is therefore particularly addressed hereinafter. Hence, the classification performance will be demonstrated by use of the two datasets presented above, namely the centrifugal chiller (Domain **A**) and the screw-chiller dataset (Domain **B**), the latter of which was collected as part of this dissertation. Consequently, the domain adaptation approach is performed and evaluated from Domain A to Domain B and vice versa.

In addition to the validation concept, information about the hardware and software basis for the experimental implementation of the model may also be an important factor for some researchers in terms of the interpretability of the results presented below. In general, the model training, including the parameter optimisation process, was performed on an HPE rack server with two AMD EPYC 7302 processors and Ubuntu 20.04 (Linux 5.11.0). It was furthermore implemented using Python 3.7 as well as several packages, including but not limited to Scikit-learn [201], NumPy [202], SciPy [203] and CVXOPT [204], as well as the standard Python library.

5.2 Preconditions

The following section outlines the precondition set out prior to the model training phase. In reference to the data source presented above, the considered feature space is summarised and an overview about the chosen evaluation metrics is given. Finally, the parameter optimisation approach is described in greater detail.

5.2.1 Shared Feature Space

The shared feature space consists of both features directly acquired from the control system during the data collection phase and the ones computed for thermodynamic state comparison, which were already in Table 4.3. As aforementioned, the feature space of Domain A and Domain B is initially not equal due to the respective chiller design constraints, which, however, represents a challenge one has to deal with in practical applications too. Therefore, only those features are selected to build up the overall dataset that represent the same physical quantities across both domains.

Recall that specific domain knowledge is not expected to be available for selecting fault indicative features to train the FDD classification algorithms. In addition, as the target domain data remain primarily unlabelled, supervised feature selection techniques, such as ReliefF [205], may also not be applicable. It can hence be deduced that instead of trying to manually construct the corresponding feature space, the fault indicative features should be identified through the CBM model. As a result, a specific feature pre-selection is neglected in this work, as it requires expert knowledge of the respective specific domain. Yet, some obvious fault unrelated indications are dismissed from the dataset such as, for example, the ambient temperatures $T_{amb,i}$ and $T_{amb,o}$. Moreover, the measured water-flow rates over the evaporator $\dot{V}_E$ and condenser $\dot{V}_C$ are not included, because the faults RVE and RVC could otherwise easily be detected and the associated sensors may have a non-negligible cost impact [137] in practice. Appendix A.5 provides a detailed overview of the features exploited throughout the model's validation phase.

For the experiments described below, $n_s = n_t = 1500$ observations from both domains were randomly selected so that the classes were represented in a balanced manner to form the training dataset. Furthermore, additional 500 observations were extracted from the target domain, which is referred to as the testing dataset in the remainder of this chapter. Both datasets were previously balanced, i.e. all classes were ensured to be represented by an approximately equal number of observations. This step was taken to avoid the model from being skewed towards one or multiple majority classes and to give all classes the same priority throughout the model training.

However, since it is expected that unlabelled observations occur more frequently than the ones being assigned to the normal class, the imbalance ratio of the test dataset validating the fault detection algorithm cannot be disregarded. A corresponding analysis with respect to the number of available labels will therefore be performed in Section 5.3.1. Another important point must be emphasised in this context. Although both domains are fully labelled in their original form, the labels of the target domain are only exploited to a full extent for testing the final CBM model classification performance.

5.2.2 Metrics

To evaluate the overall CBM model classification performance, as well as for each of the FDD layers, appropriate metrics must be defined to allow for proper comparison. These metrics are applied to numerically indicate the model's operational capability and to demonstrate its effectiveness by observing the true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) of the test dataset classified by the proposed model and the comparative algorithms. With respect to previous studies closely related to this work [6], [74], [87], [91], [132], [138], [139], [142] the performance metrics listed in Table 5.1 are introduced to allow for appropriate comparability. While most of these

Table 5.1: Performance metrics applied for model comparison.

Designation	Definition/Abbreviation
True positive rate, sensitivity, recall	$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
True negative rate, specificity	$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN+FP}}$
False negative rate	$\mathrm{FNR} = 1 - \mathrm{TPR}$
False positive rate	$\mathrm{FPR} = 1 - \mathrm{TNR}$
Accuracy	$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{FP} + \mathrm{TP} + \mathrm{FN} + \mathrm{TN}}$
Receiver operating characteristic	ROC
Area under the (ROC) curve	AUC
Matthews correlation coefficient	$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \cdot (\mathrm{TP} + \mathrm{FN}) \cdot (\mathrm{TN} + \mathrm{FP}) \cdot (\mathrm{TN} + \mathrm{FN})}}$

indications aim at explaining the overall algorithmic performance via a single numerical value, the ROC rather describes the model performance at varying decision thresholds. In this dissertation it is particularly used to compare the discriminatory capabilities [91] between the fault detection algorithm and the respective baseline models presented below. This is mainly due to the fact that this type of performance representation illustrates the entire range of threshold variability associated with a binary classifier, and thus the trade-off between the TPR and the FPR [153, p. 261]. In this way, the model can be evaluated independently of the threshold, which may better reflect the overall suitability for the given classification task. To represent the ROC curve as a single numerical value, the AUC is commonly applied, whereby a value of 1 marks the best possible performance for the specific problem and 0.5 could represent a random classifier.

A critical factor in the model validation is the class imbalance ratio imposed by the test dataset, which cannot be disregarded and for which an appropriate metric must be introduced accordingly. Since the test data is balanced in terms of the available classes, i.e. the fault-free and all fault classes, it becomes an imbalanced problem with respect to the fault detection layer. This leads to a problem in that the validity of many metrics, such as accuracy, is diminished with respect to the evaluation of the model's classification ability. In fact, most metrics tend to indicate an overoptimistic estimation of the classifier ability on the majority class [206]. According to Chicco et al. [207], two evaluation metrics are alternatively applied in this regard, namely F-score and Matthews correlation coefficient (MCC), the latter of which should be preferred for binary classification tasks. The reason for this is twofold: first, MCC provides an unbiased metric [208] that is invariant to class swapping and, second, it incorporates both the number of positive and negative samples being correctly classified.

5.2.3 Parameter Optimisation

One of the key challenges when dealing with ML algorithms arise from determination of the respective hyperparameters that one can use to control their behaviour [200, p. 117],

for which appropriate search strategies must be employed. With regards to the proposed CBM model, the parameter optimisation processes of the fault detection and the fault isolation algorithm are performed successively.

As illustrated in Figure 5.2, the hyperparameter search space is first narrowed down by manually defining search boundaries from which thereafter an appropriate discretisation is generated. This process is widely known as grid-search and is often the preferred search method in conjunction with SVMs [186]. To prevent the model from overfitting [153, p. 324], k-fold cross-validation is often used to define a suitable subset of hyperparameters [141, p. 217] and is also applied in this work during the parameter tuning process. Recall from Section 3.5.3 that the evaluation metric (20) is applied as an performance indicator as proposed by Liu et al. [157] in the context of PU learning. Another parameter affects the OC-SVM, namely the Gaussian width parameter γ_{ocsvm}. Following the approach presented in [91], this parameter is tuned using the heuristic proposed by Wang et al. [167], the so-called tightness detection algorithm.

After an appropriate subset of hyperparameters associated with the fault detection model could be identified, the second FDD layer needs to be taken into account. In fact, the parameter optimisation process is fairly similar to the former one and, yet, some differences must be highlighted. As shown in Figure 5.3, both the domain adaptation procedure and the reduction of the process variability are first conducted. Then, the parameter optimisation process implies k-fold cross-validation on the fault patterns originating from the source domain. Thereby, an essential difference to the previous layer arises.

As a result of the feature transformation and feature selection steps described in Section 3.6, both the domain adaptation and the reduction of process variability is conducted, while the parameter optimisation problem reduces as an ordinary supervised multi-class ML case. In this way, the source domain, including its labels, is fully exploited throughout the hyperparameter optimisation process, whereby common performance metrics are eligible to be used to indicate an appropriate parameter subset. This work applies ACC in the following. So far, the hyperparameter optimisation process has only been separately described for the fault detection and the fault isolation algorithm. The practical

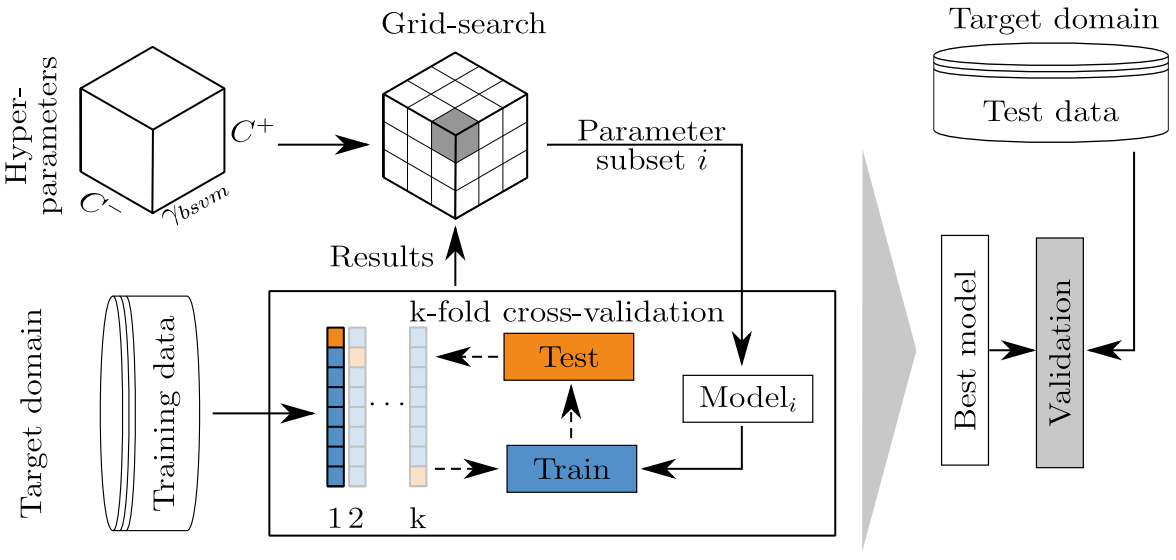

Figure 5.2: Hyperparameter optimisation of the fault detection model.

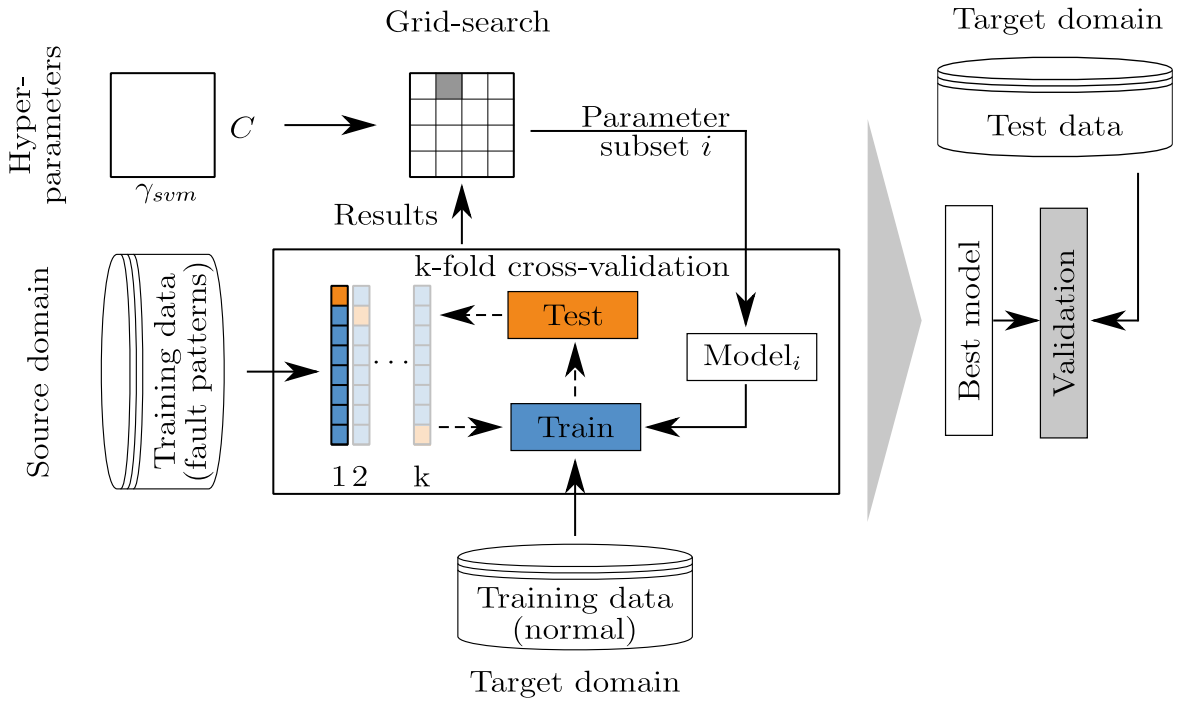

Figure 5.3: Hyperparameter optimisation of the fault isolation model. Target domain training data associated with the normal operating condition are used for domain adaptation purposes.

relevance, however, results from the combination of two, which ultimately influences the overall applicability of the model, a case that will be specifically addressed in Section 5.4. Accordingly, the two layers within the CBM model are optimised consecutively. For more information about this procedure, see Section 3.1.3. Furthermore, it is worth noting that for reasons of computational performance, 3-fold cross-validation will be employed hereinafter.

5.3 Individual FDD Results

This section provides an overview of the results obtained from the model validation phase, whereby each of the three layers of the FDD block is reviewed, separately. Following the CBM sequence presented in the beginning of this work, the fault detection ability is first discussed and the proposed PU learning approach is compared with the two state-of-the-art baseline models. Subsequently, the results associated with the fault isolation layer are presented with a special view on the domain adaptation procedure. The section then closes with the results of the fault identification phase and thus with the estimation of the health index, i.e. the fault severity.

5.3.1 Novelty Detection

To demonstrate the classification ability of the BSVM classifier, it is compared with the two baseline algorithms identified in Section 2.6, both of which allow novelty detection tasks to be performed by solely exploiting partially labelled data in the model training phase. Thus, the models proposed in [6] and [91] will be employed in the following

comparison, whereby the former applied squared prediction error (SPE) indices to detect chiller faults while the latter utilises a OC-SVM. The fact that all three models are trained using the PCA-transformed residual subspace allows for a reasonable comparison of the approaches. Figure 5.4 well illustrates the advantage of this method, namely the striking occurrence of fault patterns.

Even though all three models rely on the reduction of process variability, they differ in the number of discarded PCs (k_{pc}). Other hyperparameters worth mentioning in this context are the BSVM parameters, whose search space can be found in Table 5.2. Note that only the results associated with the highest scored parameter subset are presented in the figures below.

Besides, recall from Hypothesis H3.1 presented in Section 2.6 that only a minor number of labelled observations are assumed to be available from the target domain, while the rest of the observations remain unlabelled. To study the effect of the proportion of labelled normal data being available on the model's classification performance, a new parameter θ is introduced in the remainder of this section to represent the proportion of labelled samples within the training dataset. For example, $\theta = 0.1$ refers to 10 % of the data associated with the normal class being actually labelled ($y_{t_i} = 1$), whereas the remaining 90 % are assumed to be unlabelled ($y_{t_i} = -1$).

As mentioned earlier, one crucial parameter for successfully reducing the process variability via PCA while maintaining an adequate number of fault-indicating features is the CPV described by the discarded PCs. Consequently, Figure 5.5 illustrates the achieved MCC scores of the three models for varying settings of k_{pc} at $\theta = 0.2$. As the figure shows, the BSVM algorithm was able to achieve the highest scores across both domains. Additionally, it proves to be less dependant on k_{pc} with regards to its classification per-

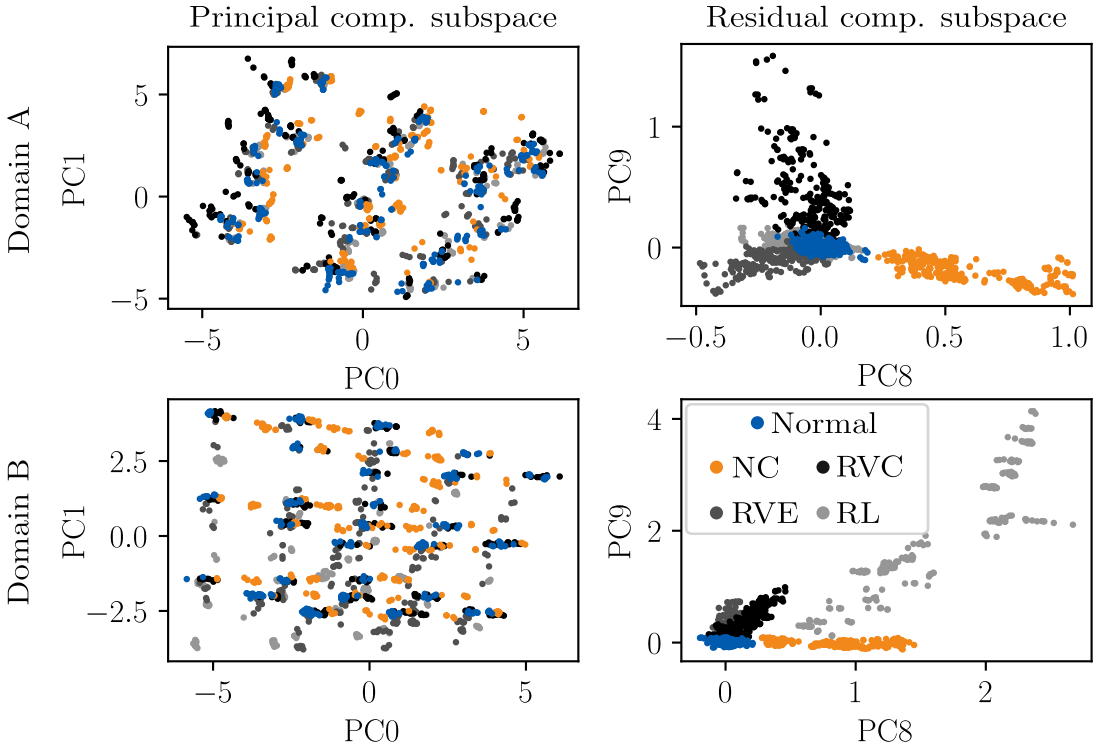

Figure 5.4: Examples of principal component and residual component subspaces represented by the training dataset.

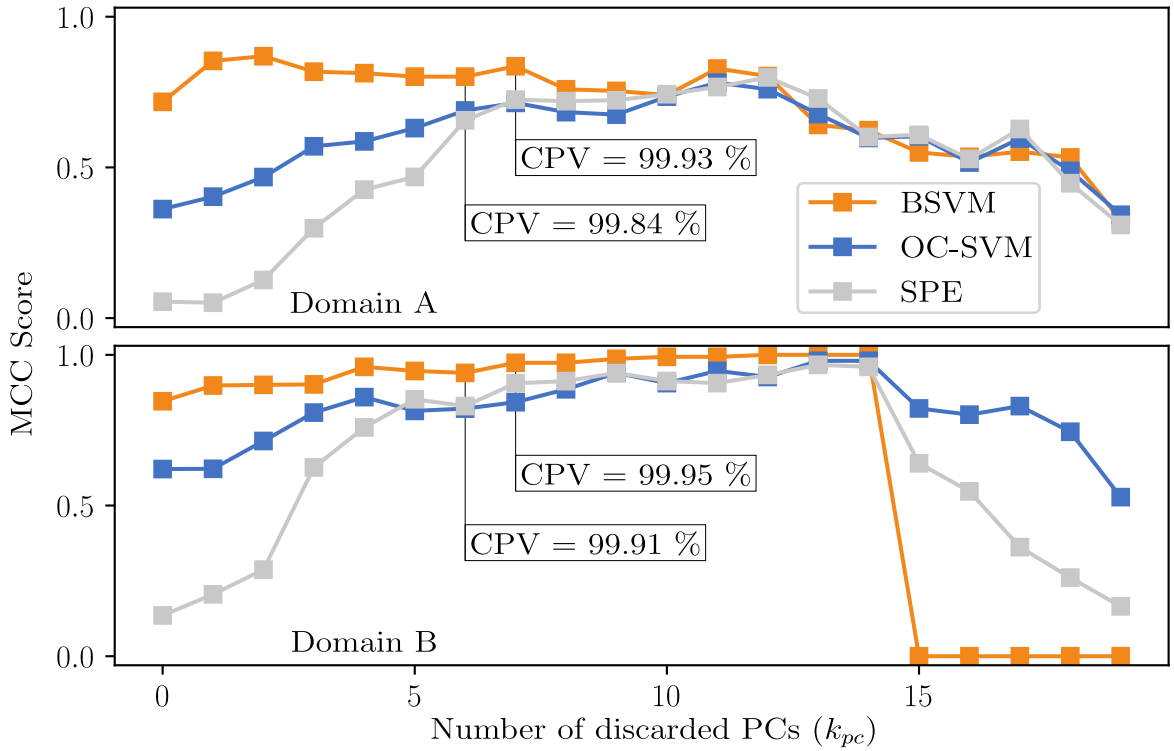

Figure 5.5: MCC Scores as functions of discarded PCs for $\theta = 0.2$ of (upper) Domain A and (lower) Domain B.

formance. This is a decisive advantage because this parameter can hardly be, if at all, estimated in practice due to the unavailability of labelled target domain fault patterns.

It becomes apparent that both the OC-SVM and the SPE approach strikingly tend to yield degraded fault detection performance for a low CPV, which is especially pronounced for the latter model. The figure additionally shows the importance of the threshold parameter CPV_{kpc}, as too low values preserves the majority of the process variability in the training dataset while too high values could lead to the exclusion of crucial fault characteristic features. In both situations, however, the data classification leads to poor results and, in some cases, may even prevent the fault detection algorithm from providing its intended functionality. This becomes especially obvious for Domain B for $k_{pc} \geq 15$, where the remaining features may not allow for an appropriate recovery of a decision boundary, which explains the rapid drop of the BSVM performance. For the problem at hand, Figure 5.5 indicates $CPV_{kpc} = 99.9\ \%$ to be an appropriate threshold. For reasons of consistency, the number of discarded features is set to $k_{pc} = 7$ for both datasets in the following.

Another aspect concerns the amount of labelled data being available at training time. Starting from the previously introduced parameter θ, Figure 5.6 shows the achieved MCC scores of the proposed approach as well as those of the two baseline models. From this it can be deduced that the BSVM algorithm shows the higher performance compared to the other two with the only exception being $\theta = 0.1$, where the classification ability appears to be lower. This indicates that a meaningful ratio between the labelled and unlabelled data should be present to enable the algorithm to well approximate the fault discriminative structure. Interestingly, the two baseline models perform nearly equally well across both domains. However, the OC-SVM model proves to be less influenced by

Table 5.2: The BSVM search space and selected parameters.

Parameter designation	Search space	Selected parameters	
		Domain A	Domain B
γ_{bsvm}	$2^{-10}, 2^{-9}, \ldots, 2^{4}$	2^{-3}	2^{1}
C^{+}	$10^{0}, 10^{1}, \ldots, 10^{7}$	10^{6}	10^{3}
C^{-}	$10^{0}, 10^{1}, \ldots, 10^{7}$	10^{4}	10^{2}

k_{pc} and is therefore applied in this work in the case the model must deal with unseen data patterns. Another useful property of this algorithm concerns the fault identification layer, namely the estimation of the health index, which will be demonstrated later.

Alternatively, the SPE approach can be used for fault detection as it offers distinct advantages in terms of the hyperparmeter tuning process, while heuristic and meta-heuristic search strategies must be employed for the OC-SVM and BSVM algorithms, respectively. Accordingly, the highest scored parameter subset of the BSVM hyperparameters for $k_{pc} = 7$ and $\theta = 0.2$ are set out in Table 5.2, which are used for the comparison described hereinafter.

To demonstrate the discriminating power of the proposed fault detection algorithm, ROC analysis is carried out in this work, as the eponymous metric conceptually allows conclusions to be drawn about the classification ability for varying thresholds. According to Fawcett [209], ROC curves are moreover insensitive to changes in the class distribution and are thus well suitable for evaluating the classification performance by use of an imbalanced test dataset. Likewise, they are invariant to class swapping, since this only leads to a mirroring of the curve along the descending diagonal axis [210]. Figure 5.7 illustrates the individual class performance of the considered fault detection algorithms, each of which

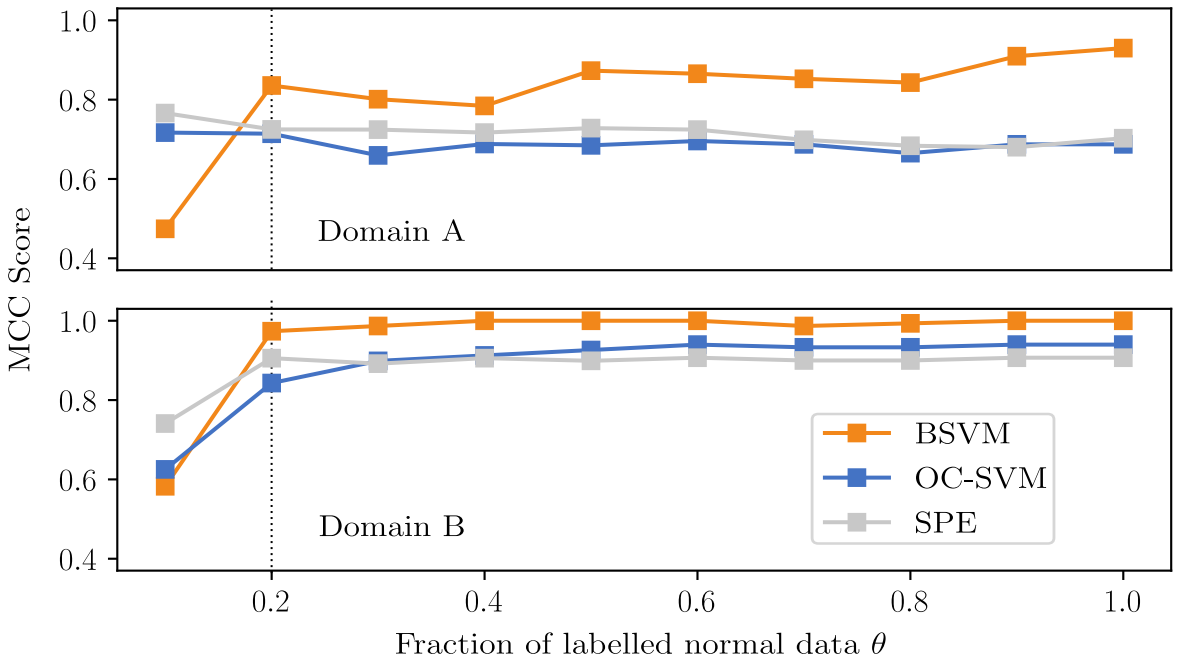

Figure 5.6: MCC Scores as functions of θ of (upper) Domain A and (lower) Domain B both at a reduced of $CPV \approx 99.9\%$ with $k_{pc} = 7$.

is evaluated at SL=1. It should be noted that in the case of the SVM-based approaches, the respective class scores used to derive the ROC scores are related to the distances from the separating hyperplane in feature space, whereas the scores of the SPE represents the Euclidean distance from the origin computed in the RCS. For more information on the latter, interested readers are referred to [6] and [211].

The figure also shows the differences between the two datasets, with the fault discrimination capabilities varying due to the different distributions underlying the two datasets. For example, RL appears to be hardly distinguishable from a normal chiller operational state in Domain A, whereas Domain B shows a pronounced indication of this fault pattern. On the other hand, the fault NC seems to be detectable with little effort across both domains as a result of its distinctive fault characteristics, confirming the findings presented in [133]. Note that further ROC curves as well as the associated AUC scores for all considered severity levels are presented in Appendix A.6.

The ROC analysis in summary reveals the discriminative power of the investigated classifiers and demonstrates improved classification performance through PU learning for fault detection tasks. In particular, the BSVM classifier provides the highest classification performance. However, although being on a very comparable scale for the presented setting of k_{pc} and θ, the performance gap between the baseline models and the BSVM-based approach appears to be higher for other parameter settings. It can therefore be concluded that, although all models confirm Hypothesis H3.1, an improved fault detection algorithm can be obtained via PU learning in comparison to density estimation and is rather insensitive to the hyperparameter setting. Consequently, this justifies the correctness of Hypothesis H2.2. Yet, it is important to emphasise that the properties of the BSVM

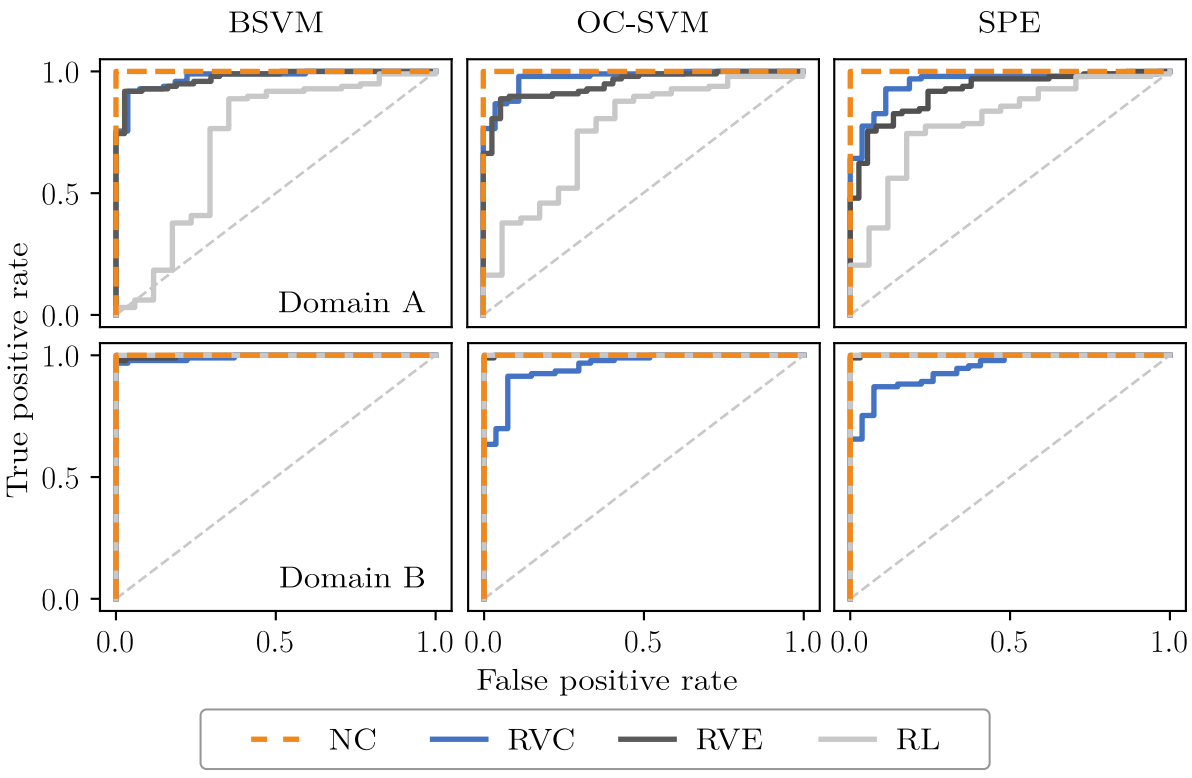

Figure 5.7: Discriminatory power of the compared novelty detection algorithms represented by ROC curves for each fault type in both domains at SL=1.

model can only be taken advantage of in case that certain fault patterns are already contained in the target domain training dataset, even if only represented by unlabelled observations. If this condition is not satisfied, then the classification performance of the proposed fault detection algorithm reduces to that of the OC-SVM-based approach.

This analysis furthermore indicates the correctness of a key assumption introduced in Section 3.7. That is the existence of a dependency between the fault severity and the distance from the SVM decision boundary as determined through its training process. This creates the basis for the fault identification layer, the results of which are further elaborated in Section 5.3.3.

5.3.2 Fault Classification

In the fault detection layer, one relies only on data originating from the target domain and therefore the heterogeneity or, more precisely, the domain discrepancy between systems is not necessarily subject to consideration. This, however, changes in the development of the fault isolation layer, as the domain adaptation procedure forms an integral part in terms of improving the overall classification ability, which is hereinafter discussed.

Comparison of Domain Adaptation Models

To demonstrate the effectiveness of the presented model, the PCA-based approaches presented in Section 3.6.2 as well as the state-of-the-art approaches, also mentioned earlier, are implemented and their results are comparatively presented. As a reminder, the latter concerns the three domain adaptation models TCA [180], CORAL [181] and SA [182], leaving aside the methods based on deep neural networks for computational reasons and, above all, because of the problem of tuning the associated hyperparameters.

It is important to note that, based on the preliminary considerations, PCA is at the very heart of the comparison presented below. This not only affects the presented approach but rather the implementation of all models. As has been argued and shown previously, fault patterns are especially pronounced in the RCS and, thus, the reduction of the process variability has a particularly advantageous effect on the overall classification performance. Therefore, to achieve comparable results among the various domain adaptation methods, PCA has also been introduced as a preprocessing step to improve the overall fault sensitivity for the domain adaptation methods TCA and CORAL. To this end, the decomposition of the design matrices is first separately performed on the source and the target domain data. In a second step, the resulting residual components are mapped back to the original feature space before performing the domain adaptation process using the relation (6). As will be demonstrated subsequently, this measure avoids training the SVM classifier by use of highly differing feature representations.

Furthermore, the number of discarded PCs by means of the parameter k_{pc} is crucial. In fact, if $k_{pc} = 0$ there is no decomposition and domain adaptation is performed on the available fault patterns by use of entire input space. For all values $k_{pc} > 0$, however, only the resulting residual subspaces are aligned, which is expected to yield comparatively better results. Furthermore, it is important to emphasise that the previously introduced parameter θ is set to 1 throughout this section. Nonetheless, its impact on the overall CBM model performance will be reviewed later.

The state-of-the-art model being closely related to the presented approach is SA. For reasons of comparability, SA is similarly implemented as the PCA based methods. To this

Table 5.3: SVM search space and selected parameters for PCA(X_t) and $k_{pc} = 7$.

Parameter designation	Search space	Selected parameters	
		Domain A → B	Domain B → A
γ_{svm}	$2^{-10}, 2^{-9}, \ldots, 2^{10}$	2^{-9}	2^{-5}
C	$10^{-10}, 10^{-9}, \ldots, 10^{10}$	10^{4}	10^{1}

end, the PCA(X_s), PCA(X_t) model is first applied followed by a mapping of the resulting features to the target domain RCS. For more details on the latter, see the original paper [182]. As a result, SA can be implemented independently of the availability of target domain fault patterns and thus differs from TCA and CORAL.

Besides, these two domain adaptation methods also rely on further hyperparameters to be tuned, which prove to be difficult in unsupervised domain adaptation problems due to the general unavailability of target domain labels [178]. Based on this premise, TCA is only applied using the linear kernel, since the use of other kernel functions, such as the RBF, introduces further hyperparameters and leaves open the question of how these can be determined without the use of target domain labels.

Nevertheless, TCA and CORAL require a set of hyperparameters to be determined. While in the first method the number of transfer components k_{tca} to be chosen is crucial, in the second method the regularisation parameter λ must be specified, which affects both the whitening of the source domain data and the re-colouring with the target domain covariance matrix. Accordingly, the highest scored parameters can be found in Table 5.5. Note that the associated search spaces result from the original considerations in [180] and [181]. Other parameters directly affect the SVM classifier, namely C and γ (only for the RBF kernel function), the search spaces of which are listed in Table 5.3. Again, only the accuracy of the highest scored parameter subsets is presented hereinafter.

Table 5.4 shows the achieved accuracy scores in dependency on the number of disregarded principal components, the search space of which is set to $k_{pc} \in \{0, 3, 5, 7, 9\}$. The first obvious characteristic arises from the significant increase in the classification performance of the classifier trained using the residual subspace components instead of the original input space. This indicates a better differentiation of fault patterns in the RCS as a consequence of the lower discrepancy between the domains. Secondly, in view of the PCA based methods, both the PCA(X_s, X_t) and PCA(X_t) models showed improved results compared to PCA(X_s), PCA(X_t) and PCA(X_s). In particular, PCA(X_t) demonstrates somewhat more stable results for both tasks Domain A → B and Domain B → A as well as with respect to the kernel functions. Moreover, it is clear to see that training the SVM on two different feature subspaces obtained from the PCA(X_s), PCA(X_t) model may not represent an adequate choice.

In addition, the results of the proposed fault isolation algorithm and those of the state-of-the-art domain adaptation methods are of a comparable order of magnitude, with CORAL achieving the highest accuracy of all experiments. Yet, it must be noted that it is based on the assumption that certain fault patterns are equally present in the training data of both domains, which can hardly be guaranteed. Similarly, the solution obtained via TCA may be less suitable in this respect, which will be further elaborated throughout this section. From the opposing point of view, SA could represent a feasible domain adaptation approach, as it allows the residual components to be aligned and, therefore, the approach does not rely on any target domain fault patterns (neither labelled nor

Table 5.4: Classification accuracy scores in [%] of the considered domain adaptation approaches for comparison.

Model	SVM kernel	$k_{pc} =$	Domain A → B					Domain B → A				
			0	3	5	7	9	0	3	5	7	9
PCA(X_s), [1]	Linear		29.2	25.8	25.3	25.0	21.8	27.6	27.4	25.4	22.6	20.4
PCA(X_t)	RBF		28.5	15.7	20.4	30.2	13.3	25.9	27.4	21.1	15.9	12.4
PCA(X_s), [2]	Linear		62.7	85.3	81.3	76.4	90.4	61.4	71.1	73.1	73.1	47.5
PCA(X_t)	RBF		66.8	78.9	69.3	68.1	33.9	60.2	70.9	72.1	45.0	43.3
PCA(X_s)	Linear		66.6	76.4	73.7	72.0	61.7	60.9	58.7	55.5	52.2	55.0
	RBF		68.3	65.1	48.9	48.2	23.3	59.5	57.0	30.1	23.9	28.6
PCA(X_s, X_t)	Linear		66.6	87.7	95.6	95.8	**97.3**	60.9	75.4	78.9	74.6	77.6
	RBF		68.3	77.1	95.3	58.7	41.3	59.5	76.4	79.4	76.4	75.6
PCA(X_t)	Linear		66.6	75.9	79.1	78.9	82.8	60.9	75.1	81.3	81.8	53.2
	RBF		68.3	85.0	91.9	86.5	88.9	59.5	75.1	77.6	**83.8**	46.8
TCA	Linear		73.2	70.3	96.6	99.3	85.3	60.4	66.2	73.1	73.9	47.3
	RBF		24.6	73.7	85.3	44.7	49.1	26.6	65.4	68.2	70.6	46.0
CORAL	Linear		66.6	85.3	81.3	98.8	**99.5**	68.4	73.4	76.1	76.4	50.7
	RBF		71.3	86.5	87.5	77.6	43.7	67.4	74.6	**76.9**	76.4	48.8
SA	Linear		60.0	80.6	74.9	75.2	75.2	52.7	68.9	70.6	73.1	56.7
	RBF		60.2	74.0	70.3	49.1	43.2	51.7	69.9	72.9	41.8	51.0

[1] SVM trained by use of residual components
[2] SVM trained by use of residual components mapped to the original feature space

unlabelled) in the training phase. Nonetheless, as can be seen from the results above, it yields lower classification results compared to the proposed algorithm.

Interestingly, the linear kernel function seems to produce better results across the examined models, with the only exception being the proposed one where the RBF kernel function demonstrates to be the better choice. Even though some models considered in this comparison may not be feasible for fault isolation tasks in practice, two important statements can be derived: The fault isolation models aiming to first reduce the domain discrepancy before training the classifier yield higher and more reliable classification performance and, therefore, Hypothesis H1.2 can be regarded as correct. Furthermore, it shows that partially labelled target domain data can be utilised to reduce the domain discrepancy by specifically emphasising fault indicating features and, thus, proves Hypothesis H2.1 to be also correct.

Table 5.5: The hyperparameter search space and the highest scored parameters applied for comparison of TCA and CORAL.

Parameter designation	SVM kernel	Search space	Selected parameters	
			Domain A → B	Domain B → A
k_{tca}	Linear	$10, 20, 30$	10	10
	RBF		10	10
λ	Linear	$10^{-3}, 10^{-2}, \ldots, 10^{1}$	1	10
	RBF		10	10

Residual Component Subspace Dimensionality

Another important factor is the sensitivity of the fault isolation model with respect to the number of discarded principal components represented by the hyperparameter k_{pc}, which should be low to ensure its practical feasibility. This is essentially due to the fact that the optimisation of this parameter requires labelled fault related observations from the target domain, which, as discussed earlier, are not expected to be available. Although CPV seems to be an appropriate indicator for the choice of k_{pc}, it may still be subject to different results. Due to many problem-specific factors, such as the available feature space, the types of faults observed, or simply the data basis, this can potentially lead to different parameter settings that are considered appropriate. As shown in Table 5.4, the proposed model reveals decent properties in this respect. Likewise, Figure 5.8 illustrates the dependency of k_{pc} on the classification performance. From this, it becomes evident that the PCA(X_t)-based domain adaptation approach is robust with respect to the number of discarded principal components, a property that is particularly pronounced in task Domain A $\rightarrow$ B.

The task Domain B $\rightarrow$ A, however, shows degraded accuracy scores for higher values of k_{pc}. Similar behaviour could be observed in the previous section describing the fault detection layer, for which the reader is referred to Figure 5.5. This is clearly not a coincidence and underlines a statement introduced earlier, namely the trade-off between reducing the process variability and discarding fault-related information with respect to choosing k_{pc} being critical in practice. Thus, the conclusion follows to tend to slightly smaller values of $CPV_{k_{pc}}$ in case of doubts, where values between 99 % and 99.9 % seem reasonable. This is also consistent with the findings presented in [87] and [91]. In this study, however, $k_{pc} = 7$ is chosen, since it demonstrates to yield decent results across both domains. The associated SVM parameters defined through grid-search and 3-fold cross-validation are listed in Table 5.3. Moreover, this value is consistent with the results presented in the previous section, where the number of principal components to discard is equally set for the fault detection algorithm. Figure 5.9 illustrates the results obtained from the domain adaptation models for this parameter setting. It can be interpreted that the proposed algorithm produces stable results for both kernel functions and even outperforms the state-of-the-art domain adaptation methods for the task Domain B $\rightarrow$ A.

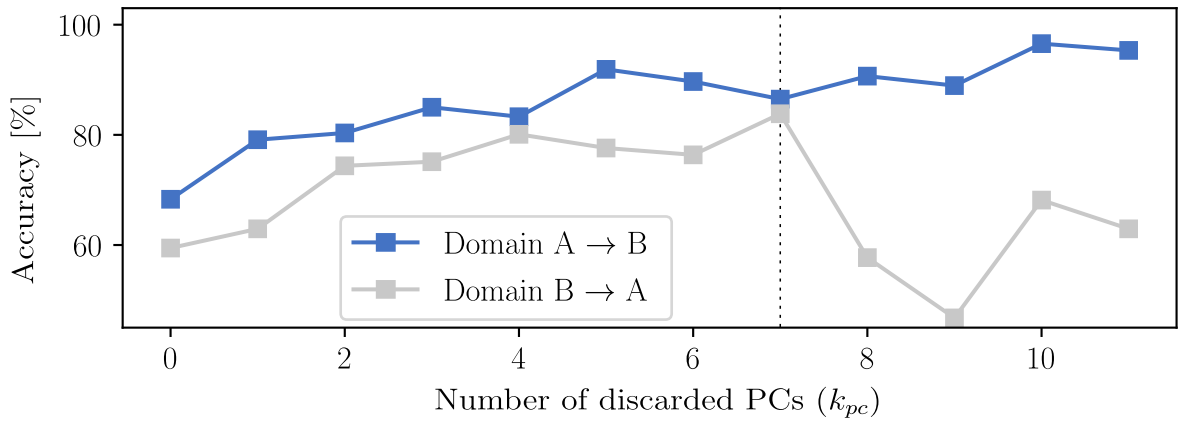

Figure 5.8: Achieved accuracy scores for both domains as functions of discarded principal components (k_{pc}).

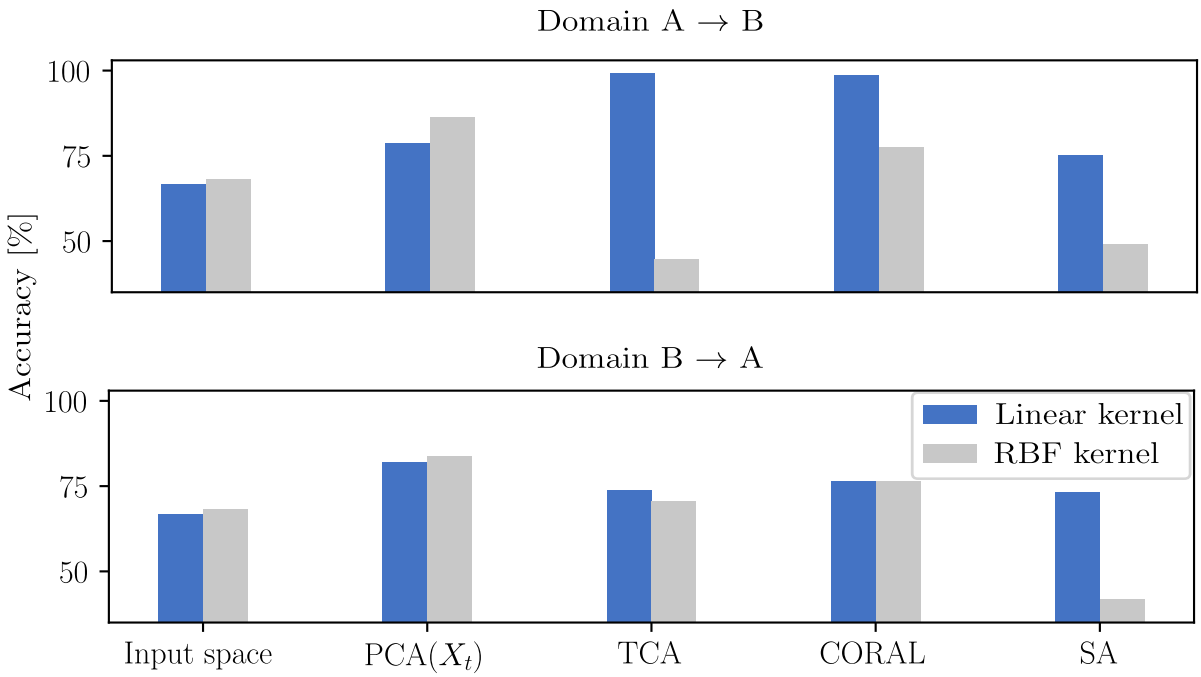

Figure 5.9: Overview of achieved accuracy scores for (far left) the original input space and the domain adaptation models at $k_{pc} = 7$.

Availability of Target Domain Fault Classes

Recall from RQ4 that a significant concern is placed on the feasibility of a domain adaptation model when the label space is not fully covered by the target domain training data, i.e. $\mathcal{Y}_t \subset \mathcal{Y}_s$. This concern, however, is often disregarded in many domain adaptation problems and the classes are assumed to be equally available in the source but also in the target domain, with the latter being unlabelled. This also applies to two of the state-of-the-art models used for comparison in this section, namely TCA and CORAL. To show the effect of neglecting this condition, a sensitivity analysis is performed with respect to the availability of (unlabelled) target domain fault classes at the model's training time.

To this end, the classifier performance for all available fault class combinations in the target domain dataset was computed. For example, at any iteration, the faults NC and RVC were selected to be available in the target domain training dataset, whereas for another iteration it was NC, RVE and RL, etc. It goes without saying that at least one fault class was present in the target domain dataset allowing to retrieve comparable solutions for all domain adaptation methods, as otherwise TCA and CORAL may not converge in the training process.

Figure 5.10 shows the average accuracy scores as well as the score distributions in form of box plots. As can be seen from this figure, the average results of TCA and CORAL are significantly reduced and their results also vary widely. Conversely, the proposed model as well as SA are designed so that they are decoupled the availability of target domain fault data, whereby the former shows significantly better results. Figure 5.11 illustrates the feature representation of the original input space and the residual subspace resulting from the proposed model. The higher dimensional feature spaces are reduced to two dimensions by performing linear discriminant analysis (LDA) on the source domain fault data. Note that only 40 % of the training data are visualised for reasons of greater

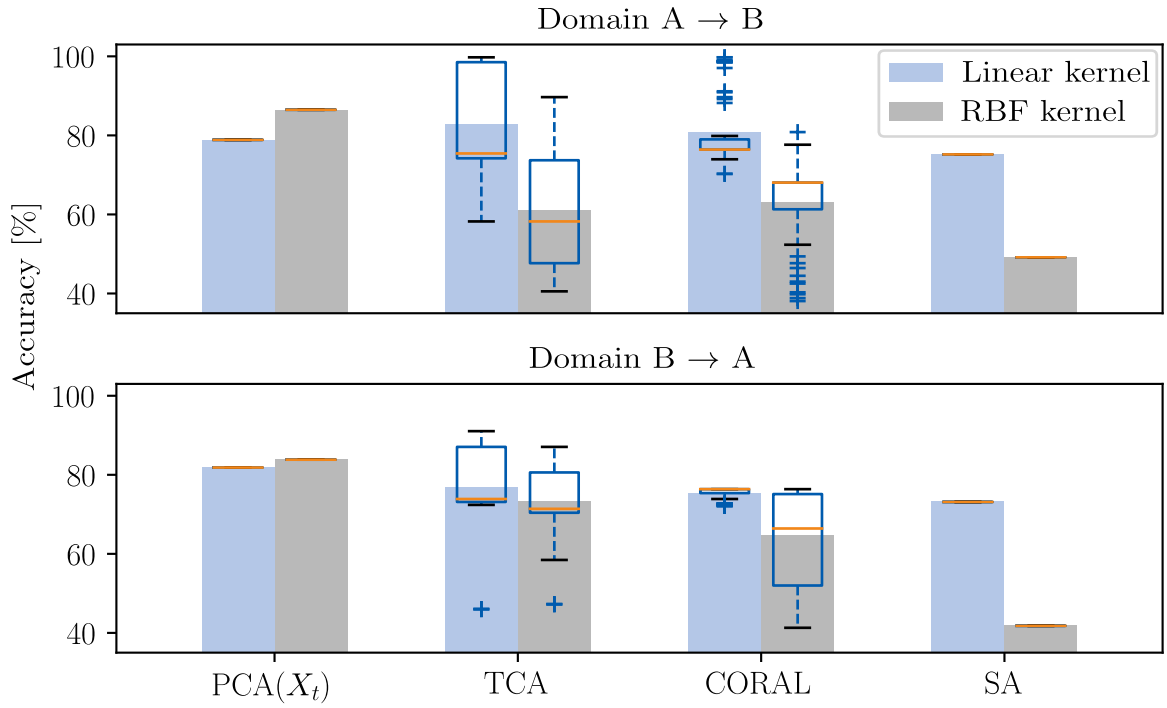

Figure 5.10: Sensitivity of domain adaptation models in dependence on the availability of fault patterns in the target domain dataset - bar plots show the average accuracy score and the box plots demonstrate the score distribution for varying fault classes combinations of the target domain being available at training time.

clarity. From the figure, one can recognise the benefits arising from the use of the residual component subspace for fault isolation tasks. This can be explained by the fact that the discriminative structure of the fault patterns appear much more pronounced and is therefore better distinguishable.

More importantly, however, the fault patterns appear to be well aligned in the RCS. In reference to Table 5.4, this fact becomes even clearer and indicates that the residual components can approximately be considered to be domain invariant features for chiller fault isolation tasks. As a consequence, this proves the correctness of both Hypothesis H4.1 and Hypothesis H4.2. More feature representations can be found in Appendix A.7.

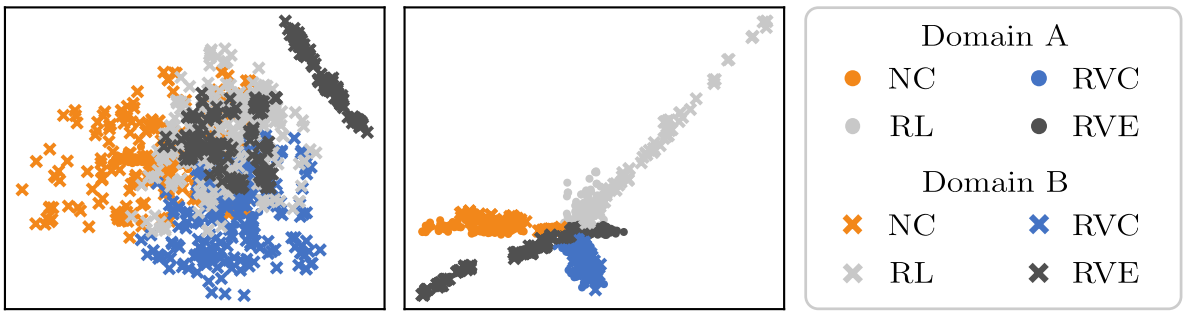

Figure 5.11: Scatter plots of the two domain for the task Domain A → B with reduced dimensionality by use of LDA for (left) the original input space and (right) the residual subspace resulting from the PCA(X_t) model.

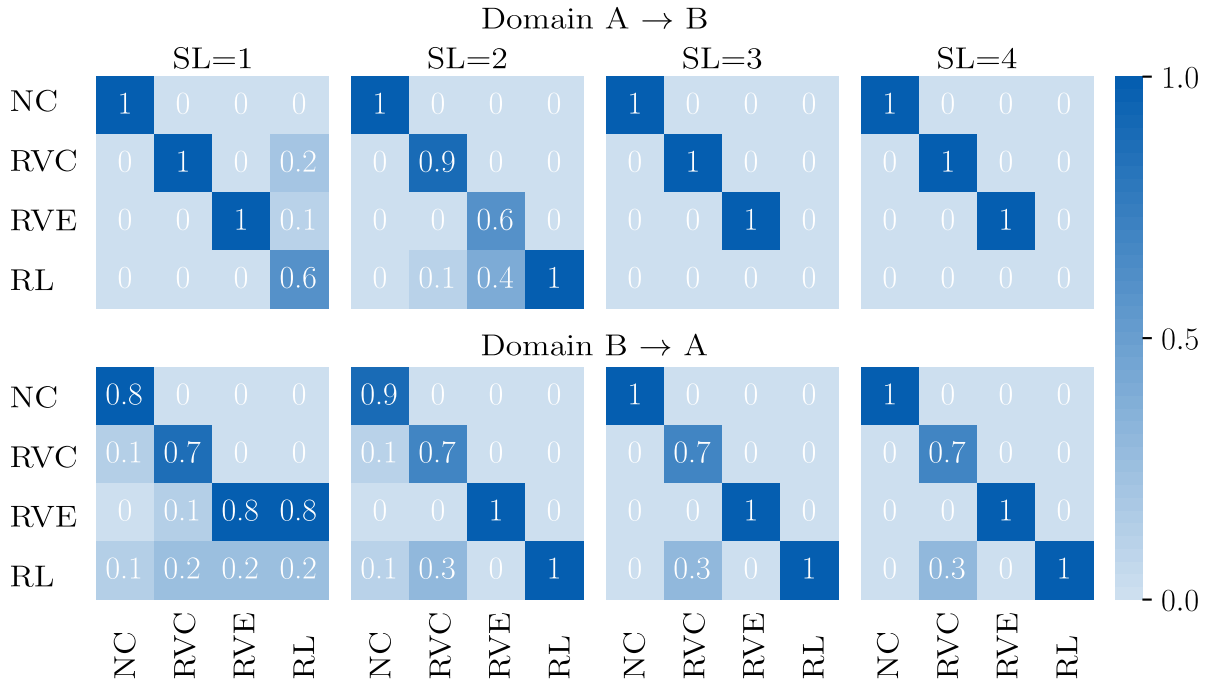

Figure 5.12: Confusion matrices determined from the proposed fault isolation model for both domains, whereby the severity levels are considered separately.

In this context it should be mentioned that these representation should be carefully interpreted, as the use of LDA for reducing the problem at hand to two dimensions may slightly distort the perception of the actual relation between the domains in the higher dimensional feature space. Nonetheless, it gives an intuition of the overall feasibility with regards to the considered domain adaptation approaches.

Performance per Severity Level

The individual class related classification performance may also be of high interest. Figure 5.12 shows the confusion matrices, which indicate the correctness of the key assumption in terms of the fault identification layer, namely the increased confidence of a fault being correctly classified for higher SL. This is described in more detail in the following chapter. Besides, the figure also indicates comparatively low classification performance on the fault RL at SL=1 for the task Domain B → A. The reason for this is twofold; first, the faults RL and RVE indeed occur with similar characteristics and, second, the fault may not be very pronounced at low SL, which can similarly be observed from Figure 5.7 in the previous section. In essence, however, it can be stated that the discriminative structure of the target domain fault patterns is well recovered from the source domain data and the proposed fault isolation approach appears to be viable in practice.

5.3.3 Health Index

The last step within the FDD block, and therefore also the last step to be taken to complete the CBM model, leads to the fault identification layer. Here, the fault severity is to be estimated, which in turn is the prerequisite for estimating RUL within the PHM

scheme. It should be emphasised again, however, that PHM is beyond the scope of this dissertation.

In general, the ISO 13374 [64] prescribes the estimation of a health index (hi), i.e. the estimation of the fault severity, to be part of the CBM procedure. As has been described previously, the application of the SVM and its extensions becomes handy in this respect. It is in this way that the decision function, i.e. before applying the sign function for the final classification task, already provides the related information. As a result, hi can be identified through the fault detection algorithm. Furthermore, by determining a reasonable scaling approach, it can also be expressed in the range $hi \in [0, 10]$ as demanded in [64].

Figure 5.13 shows the health indices obtained from the test datasets of both domains for the considered classes. As can be seen from the figure, choosing the bias terms of both the BSVM and the OC-SVM algorithm to be the lower bound to scale the respective outcome of the decision functions appears to be a practical and straightforward approach. This property is particularly convenient and therefore explains the choice of the OC-SVM instead of SPE for the handling of unknown data patterns during the model deployment phase, although both yield comparable fault detection performance rates.

Another aspect comes to the fore when taking into account the box plots obtained from the BSVM algorithm in Domain B. In fact, one can see a few outliers exceeding the lower limit of hi. In such a scenario it might be advisable to limit the index so that the

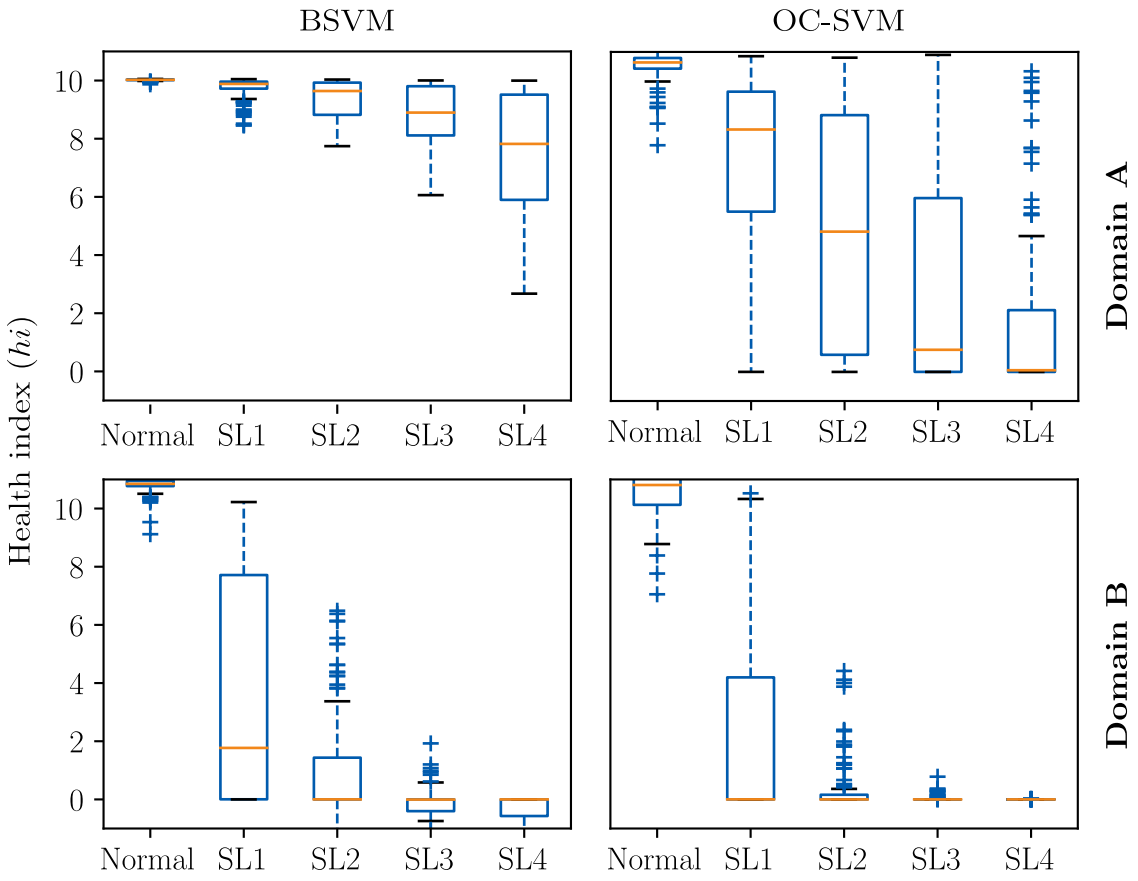

Figure 5.13: Health index estimation demonstrated by use of box plots for each considered severity level. Both classifiers are eligible for estimating hi.

lowest value possible is 0. Yet, this problem does not apply to the OC-SVM algorithm because its decision function only allows to take on the smallest value in the representation of the associated bias term. Of course, this assumption is not always true, but is valid in the case of the RBF kernel function. It goes without saying that the upper limit shall not be disregarded in this context, since any value above 10 indicates a fully functional system. Therefore, this value is also to be limited upwards.

Figure 5.13 well demonstrates the outcome of the decision function to be an eligible metric to estimate hi. Nonetheless, three aspects should be taken into account when interpreting this value: first, the fault severity estimation is highly dependent on the considered algorithm, i.e. the BSVM or the OC-SVM with regards to the presented model. Secondly, hi may not achieve comparable values when applied to different heterogeneous refrigeration systems. Ultimately, it must be underlined that according to the DIN ISO 13374 [64], $hi = 0$ shall represent a compete failure, which cannot be fully guaranteed by use of the presented fault identification approach.

In one form or another, this is yet a convenient approximation of the overall condition of a machine, which can be a good indicator of how far a particular fault has progressed. Therefore, it can be a useful metric for the subsequent PHM phase. In practice, however, one may be more interested in a trend of hi over time rather than separate estimates for each new observation recorded. This could be achieved by computing the average hi within a specified time window, which, however, is beyond the scope of this dissertation.

5.4 Holistic Model Assessment

In this section, the holistic applicability of the proposed CBM model is investigated. To this end, the overall classification performance is assessed by combining the FDD layers presented above. This is then followed by a sensitivity analysis in regards to the number of available training samples as well as an analysis concerning the stochastic robustness of the model. The section then closes with the final comments on the model's results.

5.4.1 Classification Performance

As discussed at the beginning of this work, most scientific papers on data-driven models related to industrial refrigeration systems consider their application to a single system only. In these scenarios, the problem often reduces to a supervised ML problem and can be easily dealt with. Yet, the previous sections have proven that system heterogeneity can drastically affect the classification performance. To exemplify this impact and the loss of predictive performance when system heterogeneity is neglected, the proposed model is compared to a supervised ML classifier below. For this purpose a SVM classifier is derived as a baseline model by simply training it on the source domain data in a multi-class fashion, including the normal class. To observe its classification performance, the model is then tested on the target domain test dataset. This also means that it comprises the fault detection and the fault isolation layer within one single step. Similar models were proposed, for example, in [74], [132] or [139]. It should be noted that the same boundary conditions apply as mentioned in the previous sections. Firstly, only 20 % of the normal target domain training data are expected to be accurately labelled and, consequently, θ is set to 0.2. Yet, this does not apply to the source domain and, thus, does not affect the training of the baseline model. Secondly, the SVM, but also the BSVM, search spaces presented previously are taken into account for the results presented hereinafter.

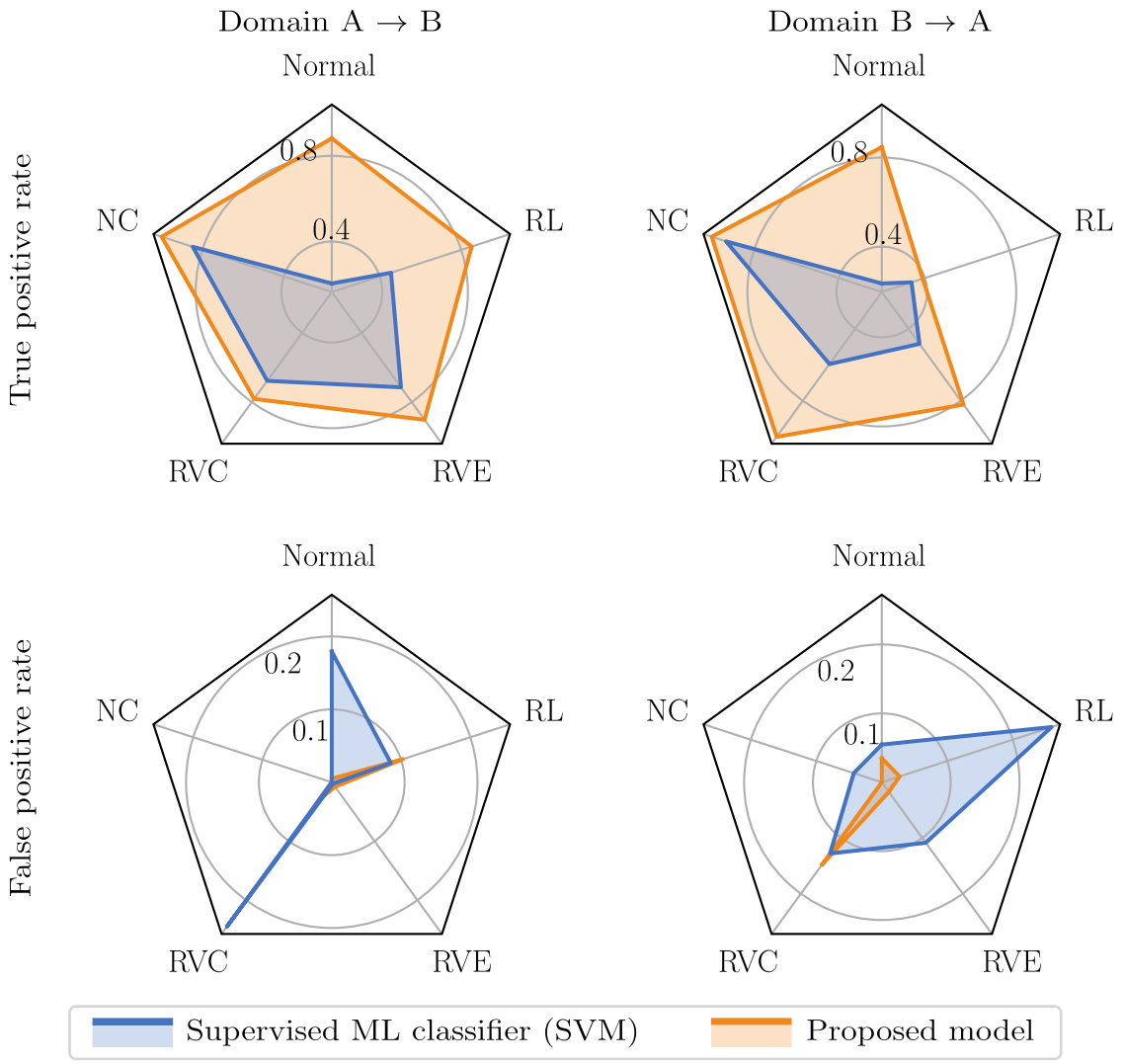

Figure 5.14: Kiviat diagrams showing the TPR and the FPR per class - the baseline model represents a classical supervised ML approach by training a SVM on the source domain data and applying it in the target domain.

Figure 5.14 shows the TPR and FPR for each class respectively with respect to the results obtained from the baseline and the proposed model. Here, attention needs first to be drawn to the normal class, i.e. the fault detection performance. Of particular note is that the baseline model appears to yield strongly degraded classification rates. This allows the assumption that the fault detection problem should be considered specific to the dedicated system. Therefore, fault detection is to be solved by incorporating data from its domain, the target domain, as in the case of the fault detection algorithm presented in this work.

This can furthermore be theoretically underlined from a thermodynamic point of view by considering the two domains presented in this work. For example, consider the evaporator design in Domain A, which is a dry expansion type. Unlike in a flooded evaporator (Domain B), the suction superheat temperature $\Delta T_{sh,suc}$ may be subject to comparatively higher fluctuations due to the varying superheat temperatures of the gaseous refrigerant and is therefore only comparable to a limited extent. This exemplifies that transferring such conditions to other heterogeneous systems may not be an adequate choice.

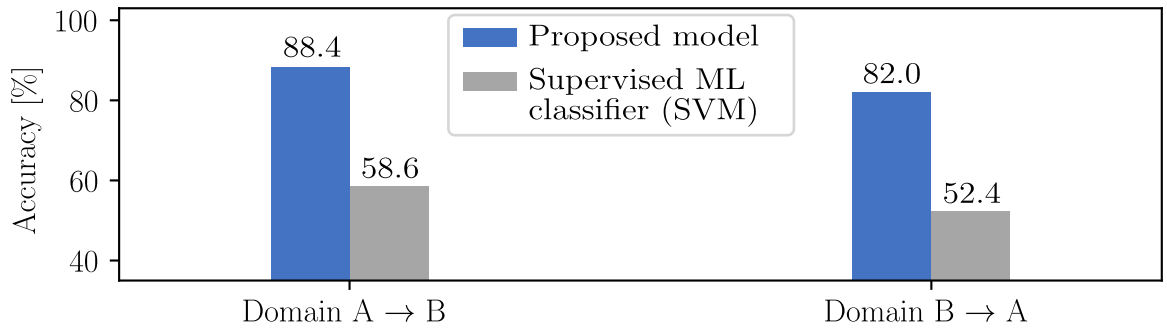

Figure 5.15: Accuracy scores of the holistic CBM model in comparison to the SVM baseline model.

The Kiviat diagrams in the figure above additionally show the superiority of the proposed model over the baseline model, which in turn emphasises once more the correctness of Hypothesis H1.2. Nevertheless, the low detection rate of RL for the task Domain B → A cannot be neglected. In fact, the poor classification scores result from both the fault detection layer as well as from the fault isolation layer. From the ROC analysis presented in Section 5.3.1, it can be seen that this fault can only be identified with comparatively low confidence. However, this is primarily the case for the first two severity levels and does not only explain the drop of the TPR. Another factor can be identified in the fault isolation layer. As can be seen from the confusion matrices shown in Figure 5.12, there are only few correctly classified observations detected for SL=1. From this, it can be inferred that the data for Domain B does not well represent the fault discriminating structure due to the smaller number of different operating states that could be observed, a condition discussed earlier in Section 4.4.5.

In addition, the TPR could be significantly improved by use of the proposed model. Similarly, the FPR could be reduced to a great extent, with the only exception being the fault RVC shown in the lower right diagram, which is slightly higher compared to the baseline model. As can be seen from the accuracy scores presented in Figure 5.15 both the presented fault detection and fault isolation approach are well suited for practical applications and surely better than following a strictly supervised ML approach. In summary, it can be stated that the proposed model outperforms the baseline model by absolute 29.8 % for the task Domain A → B and 29.6 % for the task Domain B → A.

5.4.2 Sensitivity Analysis

Despite the improved classification performance shown above compared to the supervised ML approach, the impact of the number of available training data should not be neglected. In general, low sensitivity in this regard is a desirable property in practice, as the availability of large datasets is not always guaranteed. In other words, the overall accuracy score of a CBM model should be sufficiently high, even if only a minor number of training observations can be obtained. Therefore, a sensitivity analysis is performed in this section taking into account varying numbers of training samples. Thus, the model is re-trained and tested with different values of n_s and n_t.

The model is investigated in the range $n_s = n_t \in \{300, 600, \ldots, 1500\}$, as shown in Figure 5.16. As can be seen from this illustration, both models tend to yield lower accuracy score with smaller amounts of training samples. Yet, at about 80 %, the results

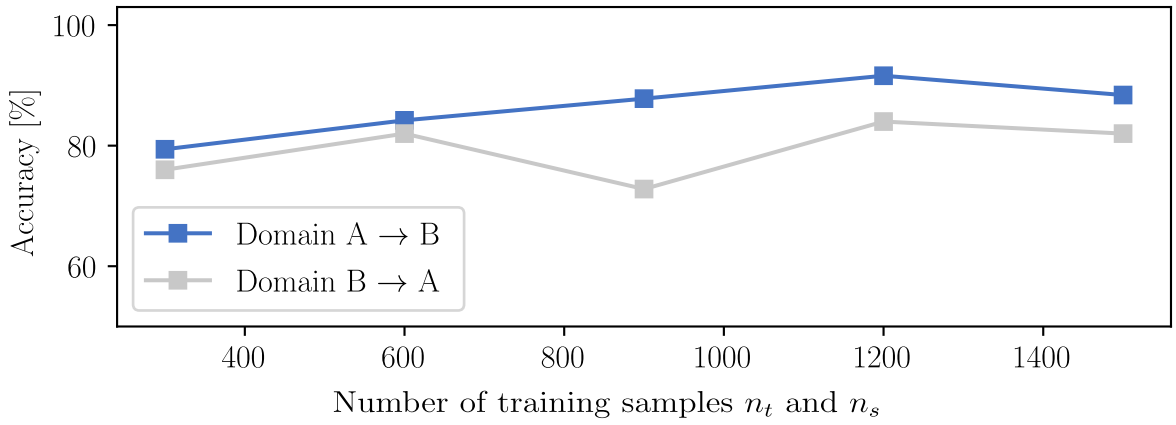

Figure 5.16: Results from the sensitivity analysis as functions of varying numbers of training samples.

obtained from both tasks still represent acceptable solutions. Interestingly, even higher scores are obtained compared to the setting described above at $n_s = n_t = 1200$. Although all experiments show acceptable performance, the lowest result can be observed for the task Domain B → A at $n_s = n_t = 900$. However, the proposed model proves to be rather insensitive to the number of training samples and can therefore be implemented even when data are scarcely available.

5.4.3 Stochastic Robustness

So far, a single dataset has been considered for both domains throughout the validation phase. However, the selection of data samples for both training and testing is subject to a stochastic process and must thus also be addressed to appropriately validate the CBM model performance. To this end, the training and validation process has been repeatability performed (12 experiments in total with $n_s = n_t = 1500$), each time with a different subset of training and testing observations being selected. It is in this way that the varying results can be observed and compared.

As shown in Figure 5.17, one can see that the increased performance of the proposed model outperforms the baseline model and achieves decent results, especially for the task

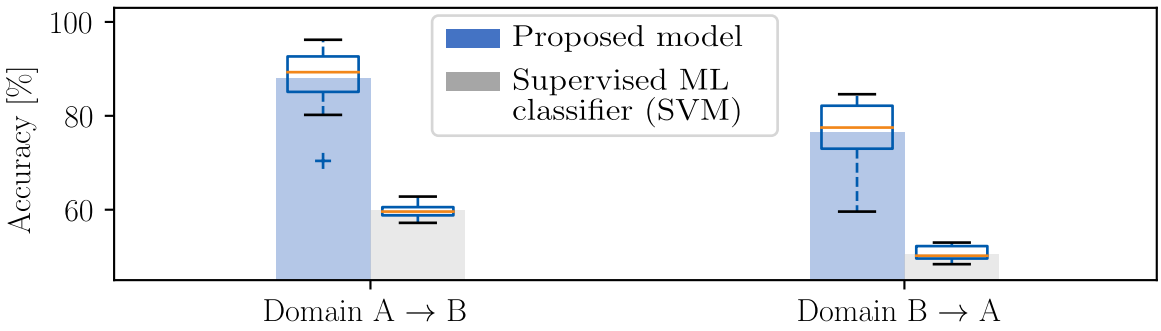

Figure 5.17: Stochastic robustness obtained by randomly selecting the training and test data from the entire dataset. The bar plots show the average accuracy score while the box plots indicate the accuracy score distribution of the experiments.

Domain A $\rightarrow$ B. The highest scores were obtained in this task with values up to 96.2 %. Yet, it must also be concluded that the fluctuations of the achieved accuracy scores are higher compared to the supervised ML approach. In fact, the fluctuations between the first and third quantile account for absolute 9.1 % (Domain A $\rightarrow$ B) and 11.1 % (Domain B $\rightarrow$ A), whereas for the baseline model it is only 3.2 % and 4.8 % respectively. In one way or another, the proposed method yields improved classification performance compared to the strictly supervised ML approach.

6 Summary and Conclusion

The core of this dissertation was the proposal of a novel data-driven predictive maintenance model (PM) for heterogeneous industrial refrigeration systems. More specifically, it dealt with a sub-process known as condition-based maintenance (CBM), the focus of which is placed on the automatic assessment of the current state of degradation of technical equipment. To achieve this, the model was developed to allow inferences about the health of the monitored system by automatically observing various quantities, primarily thermodynamic state variables, for which it applies data-driven methods from the field of machine learning.

The main contribution of this work was hence placed on avoiding time-consuming data labelling tasks and therefore promoting the development of cost effective approaches in this field. To this end, the CBM scheme was essentially abstracted to two machine learning (ML) problems, namely one-class classification and domain adaptation. While the first method was used to identify novelties that may indicate the presence of faults, the second method was used to assign the correct fault type, a process known as fault isolation. Here, the underlying idea was placed on inducing information from a fully labelled dataset, denoted as the source domain, in the model training process to apply the model to a different but related system, the target domain. The proposed CBM was additionally designed to estimate the severity of a fault, and therefore its outcome is not limited to detect faults and determining their class association, but also to provide a health index of the target system indicating the current state of degradation.

Chapter Review

The introductory chapter provided the reader a comprehensive overview of the scope of this dissertation, with particular attention to the motivation and structure of the work. Chapter 2 described the focus of this work in more detail by first outlining the scope of industrial maintenance and the economical optimisation problem arising in this context. This was then followed by the introduction of the most important terms and definitions, which mainly concerned the delimitation and categorisation of PM and CBM. For the latter, the basic classification of CBM approaches was derived, whereby the terminology with regard to data-driven methods was elucidated. Based on this, the related works were reviewed, from which the open research gap as well as the research questions driving this dissertation had be determined.

Chapter 3 then started with deriving the core assumptions for the model development, which were simultaneously formulated into a mathematical notation that was maintained throughout this work. Shortly thereafter, an abstract view of the CBM model was presented with the different layers of the information processing chain for which the corresponding algorithms are introduced. Including their respective feature engineering procedures, special attention was paid to the layers of the fault detection and diagnosis (FDD) block within the CBM scheme. The first step introduced in this context was the fault detection layer. For this purpose, the process variability contained in the data was firstly reduced by employing principal component analysis (PCA). In a second step, a one-class classifier was introduced combining both positive-unlabelled (PU) learning and density estimation. The second layer was assigned to the fault isolation problem, in which the description of the domain adaptation method was considered to be the core objective. As well as in the previous layer, PCA has also been employed for this task. Although

reducing process variability was a key aspect, minimising the domain discrepancy between the source and target domains was even more important to enable the transfer of fault related knowledge. The last layer within the FDD block constitutes the fault identification algorithm. Here, a statistical distance metric was presented to estimate the health index of the system.

The contribution of Chapter 4 was placed on the description of the data source used to validate the proposed model. This consists of two datasets, the first of which is publicly available and the second of which was collected as part of this work. Particularly in view of the latter, the data collection phase was described including the test rig design, the testing procedure as well as the examined fault types. Furthermore, the characteristic fault patterns were discussed and compared, whereby it was shown that these also vary across different domains. Yet, it was also found that similarities exist, which form the basis for knowledge transfer in the context of fault isolation.

Chapter 5 presented the overall CBM model functionality by first addressing the validation strategy as well as the metrics for quantifying its classification performance. Secondly, the fault detection ability was validated, starting with the model's sensitivity to certain parameters. It should be noted that a direct comparison was made to other state-of-the-art fault detection algorithms. Then, a receiver operating characteristic analysis has been performed to review the discriminative ability at different fault severity levels. Subsequently, the fault isolation approach was validated. Similarly, the presented approach to reduce the domain discrepancy was validated by use of three baseline domain adaptation methods. It was shown that the presented approach could yield more consistent solutions and even outperformed the baseline models in the majority of test cases. Yet, the greatest benefit in this respect arose from the model's ability to be trained without assuming the availability fault samples in the target domain training dataset, a condition that could hardly be assured in practice. This was followed by the validation of the fault identification layer, where the ability to estimate the health index was reviewed. Finally, the classification performance of the holistic CBM model was tested and the associated results were presented. Besides testing the overall accuracy, a sensitivity analysis with respect to the number of training samples and a stochastic robustness analysis were also performed.

Conclusion

Although supervised ML has been described extensively in the literature for chiller CBM tasks, it is rather limited in its broad applicability due to some constraints. In essence, when applying these types of methods, it must either be assumed that a fully labelled dataset is available for each target system or that a supervised ML model trained on data from one chiller system can simply be applied to another. Unfortunately, neither the first nor the second assumption holds in practice, as the availability of labels is scarce and the transferability of the model cannot be guaranteed due to the domain discrepancy imposed by different operating characteristics.

Overcoming this limitation by providing a transferable model available at low development cost is therefore the focus of this dissertation. As a consequence, the proposed model was developed so that it only requires a minor number of labelled data samples stemming from the target domain. Another advantage results from the fact that only one class is required to be accurately labelled, namely the normal (or fault-free) chiller operation. It is in this way that the proposed CBM model circumvents costly labelling

tasks by making use of one-class classification and domain adaptation. Accordingly, the following conclusions can be reported:

1. PCA is well suited for reducing the process variability by mapping the data into the residual component subspace (RCS) and, hence, to improve the fault detection performance. Yet, the definition of the number of principal components to be discarded (k_{pc}) remains a challenging task.

2. This study can furthermore confirm the findings of previous works, according to which discarding more than 99 % of the cumulative percent value of the explained variance associated with the normal chiller operation seems to be a reasonable choice for defining the number of features contained in the RCS.

3. By combining PU learning and density estimation, a one-class classifier can be developed, whereby the PU learning algorithm demonstrated to be less affected by k_{pc}.

4. PCA can also be employed to reduce the domain discrepancy between the source and target domain. In this work, this was archived by mapping the fault patterns of the source domain into RCS of the target domain, which is a simple but effective solution, especially from a practical point of view.

5. By use of the presented fault isolation approach, the model builds only on the fault patterns of the fully labelled source domain, while the availability of fault patterns from the target domain, even though these are unlabelled, is not required. It has been shown that this is particularly beneficial, as observing whether or not certain fault patterns are hidden within the unlabelled data of the target domain can be a tedious and potentially time-consuming task.

6. After reducing the domain discrepancy, the development of the fault isolation classifier can be treated as a normal supervised ML classifier trained by use of the source domain data. As a result, common optimisation techniques, such as grid-search or cross-validation, can be employed.

7. The health index estimation as part of the fault identification layer can be realised through the use of the support vector machine decision function. Thereby the scaling with respect to the ISO 13374 was achieved by taking advantage of the properties of the radial-basis kernel function.

In general, the proposed CBM model showed decent improvements in its classification performance in terms of its transferability to other but related heterogeneous systems. However, some drawbacks must also be discussed in this context. First, it must be noted that the use of the biased support vector machine, although it outperforms the two baseline models considered in this work, can be critical. In fact, it can only be reliably employed for fault patterns already contained in the unlabelled target domain training dataset and from whose the underlying discriminative structure can be recovered. For all other types, the problem reduces to the classification results of the presented density estimation approach.

In addition, the PU learning algorithm requires three hyperparameter to be optimised, which may have a detrimental effect on the overall training time. Another concern might be expressed by the stochastic robustness of the holistic model. As it was demonstrated

in the previous section, the model showed greater variation in its results compared to its supervised counterpart. It must be emphasised, however, that in all experiments the classification results of the supervised ML model could be exceeded by the one presented.

Recommendations

With special view to the overall applicability of the proposed CBM model, practitioners should consider the impact of seasonal trends onto its classification performance. Although the model relies only on a few labelled data samples from the target domain, they should represent a wide range of different operating conditions to achieve a low false alarm rate with respect to the fault detection layer. Additionally, one should bear in mind that in a case of emerging faults during the model deployment phase, the estimated health index may not be guaranteed to represent the actual state of the degradation. As prescribed within the ISO 13374, the lowest possible index is determined to be 0 and shall indicate a complete machine failure, which cannot be fully ensured with the presented fault identification approach.

Outlook

The most crucial aspect in this dissertation, or with data-driven approaches in general, has been proven to be the availability of sufficient amounts of data and how well the included observations model the underlying distributions. Thus, future work will be attributed to the investigation of the use of multiple source domains to develop a cross-domain fault isolation classifier. In addition, data augmentation methods will also be explored in this context to overcome the limitations imposed by the lack of data. Another aspect concerns the impact on seasonal trends on the overall classification performance, i.e. training the model by use of insufficient data from the target domain.

Ultimately, it can be assumed that the proposed CBM model, although it was validated against chiller data, may also be applicable to other sub-systems, such as air handling units, and could potentially be applied beyond the scope of industrial refrigeration. This, however, will be subject to future works.

References

[1] J. Yan, Y. Meng, L. Lu and L. Li, "Industrial big data in an Industry 4.0 environment: challenges, schemes, and applications for predictive maintenance," in *IEEE Access*, vol. 5, 2017, pp. 23 484–23 491.

[2] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using IoT sensor data," in *Proceedings of 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Bejing, China, 2017, pp. 87–90.

[3] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone and A. Beghi, "Machine learning for predictive maintenance: a multiple classifier approach," in *IEEE Transactions on Industrial Informatics*, vol. 11, 2014, pp. 812–820.

[4] S. Alaswad and Y. Xiang, "A review on condition-based maintenance optimization models for stochastically deteriorating system," *Reliability Engineering & System Safety*, vol. 157, pp. 85–101, 2017.

[5] R. Heinze, *Industrie 4.0 im internationalen Kontext: Kernkonzepte, Ergebnisse, Trends*. Beuth Verlag, 2017.

[6] A. Beghi, R. Brignoli, L. Cecchinato, G. Menegazzo, M. Rampazzo and F. Simmini, "Data-driven fault detection and diagnosis for HVAC water chillers," *Control Engineering Practice*, vol. 53, pp. 79–91, 2016.

[7] A. R. Mohanty, *Machinery Condition Monitoring: Principles and Practices*. CRC Press, 2014.

[8] M. Focke and J. Steinbeck, *Steigerung der Anlagenproduktivität durch OEE- Management: Definitionen, Vorgehen und Methoden– von manuell bis Industrie 4.0*. Springer, 2018.

[9] R. K. Mobley, *An Introduction to Predictive Maintenance*. Butterworth Heinemann, 2002.

[10] Statista. "Predictive Maintenance – Market Report 2019-2024." Martin Placek, Ed. (2019), [Online]. Available: www.statista.com/statistics/748080/global-predictive-maintenance-market-size (visited on 05/24/2020).

[11] H. Apel, *Instandhaltungs-und Servicemanagement: Systeme mit Industrie 4.0*. Carl Hanser Verlag, 2018.

[12] Statistisches Bundesamt, "Dienstleitungen Strukturerhebung im Dienstleitungsbereich Reperatur von Datenverarbeitungsgeräten und Gebrauchsgütern," Fachserie 9, 2019.

[13] S. Feldmann, R. Lässig, O. Herweg, H. Rauen and P. M. Synek, *Predictive Maintenance: Servicing tomorrow – and where we are really at today*. Roland Berger, 2017.

[14] J. Lee, E. Lapira, B. Bagheri and H. Kao, "Recent advances and trends in predictive manufacturing systems in big data environment," *Manufacturing Letters*, vol. 1, no. 1, pp. 38–41, 2013.

[15] W. Zhang, D. Yang and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2213–2227, 2019.

[16] J. Lee, C. Jin, Z. Liu and H. D. Ardakani, "Introduction to data-driven methodologies for prognostics and health management," in *Probabilistic Prognostics and Health Management of Energy Systems*, Springer, 2017, pp. 9–32.

[17] C. Anger, "Hidden semi-Markov models for predictive maintenance of rotating elements," Ph.D. dissertation, Technische Universität Darmstadt, 2018.

[18] E. Zhang, Q. Thurier and L. Boyle, "Improving clinical named-entity recognition with tranfer learning," *Connecting the System to Enhance the Practitioner and Consumer Experience in Healthcare: Selected Papers from the 26th Australian National Health Informatics Conference*, vol. 252, pp. 182–187, 2018.

[19] L. Mou, Z. Meng, R. Yan, *et al.*, "How transferable are neural networks in NLP applications?" In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 479–489.

[20] European Commission, *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions- An EU Strategy on Heating and Cooling*, 2016.

[21] A. Afshari and L. Friedrich, "A proposal to introduce tradable energy savings certificates in the emirate of Abu Dhabi," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1342–1351, 2016.

[22] T. Wilbanks, V. Bhatt, D. Bilello, *et al.*, *Effects of Climate Change on Energy Production and Use in the United States*. US Department of Energy Publications, 2008.

[23] S. Wang, Q. Zhou and F. Xiao, "A system-level fault detection and diagnosis strategy for HVAC systems involving sensor faults," *Energy and Buildings*, vol. 42, no. 4, pp. 477–490, 2010.

[24] X. Zhao, "Lab test of three fault detection and diagnostic methods' capability of diagnosing multiple simultaneous faults in chillers," *Energy and Buildings*, vol. 94, pp. 43–51, 2015.

[25] D. Sun, *Handbook of Frozen Food Processing and Packaging*. CRC Press, 2016.

[26] A. A. Minea, *Advances in New Heat Transfer Fluids: From Numerical to Experimental Techniques*. CRC Press, 2017.

[27] A. G. Mahyari, "Domain adaptation in robot fault diagnostic systems," *Engineering, Computer Science*, 2018.

[28] W. Mao, J. He and M. J. Zuo, "Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[29] Y. Fan, X. Cui, H. Han and H. Lu, "Chiller fault diagnosis with field sensors using the technology of imbalanced data," *Applied Thermal Engineering*, vol. 159, 2019.

[30] K. Yan, A. Chong and Y. Mo, "Generative adversarial network for fault detection diagnosis of chillers," *Building and Environment*, vol. 172, 2020.

[31] M. G. Pecht and M. Kang, *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*. Wiley, 2018.

[32] D. Li, Y. Zhou, G. Hu and C. J. Spanos, "Identifying unseen faults for smart buildings by incorporating expert knowledge with data," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1412–1425, 2018.

[33] P. Wang and R. Gao, "On-line fault detection and diagnosis for chiller system," in *Proceedings of the IEEE International Conference on Automation Science and Engineering*, 2016.

[34] Deutsches Institut für Normung e. V., *Condition monitoring and diagnostics of machines - vocabulary*, DIN ISO 13372: 2012(E/F) (DIN ISO), 2012.

[35] M. M. Moya and D. R. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Networks*, vol. 9, no. 3, pp. 463–474, 1996.

[36] N. Courty, R. Flamary, D. Tuia and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, 2017.

[37] Deutsches Institut für Normung e. V., *Instandhaltung - Begriffe der Instandhaltung*, DIN EN 13306: 2018-02 (DIN EN), 2018.

[38] S. O. Duffuaa and M. Ben-Daya, *Handbook of Maintenance Management and Engineering*. 2009.

[39] R. D. Cigolini, A. V. Deshmukh, L. Fedele and S. A. McComb, *Recent advances in maintenance and infrastructure management*. Springer, 2009.

[40] C. Freund, "Die Instandhaltung im Wandel," in *Instandhaltung technischer Systeme: Methoden und Werkzeuge zur Gewährleistung eines sicheren und wirtschaftlichen Anlagenbetriebs*, Springer, 2010, pp. 1–22.

[41] B. Wiegand, R. Langmaack and T. Baumgarten, *Lean Maintenance System: Instandhaltungszeit Null-volle Wertschöpfung; Workbook für Manager und Mitarbeiter in der Instandhaltung von Anlagen*. Lean Management Institut, 2005.

[42] M. P. Stephens, *Productivity and Reliability-Based Maintenance Management*. Purdue University Press, 2010.

[43] B. Leidinger, *Wertorientierte Instandhaltung: Kosten senken, Verfügbarkeit erhalten*. Springer, 2017.

[44] D. Falcone, F. D. Felice, G. D. Bona, V. Duraccio, A. Forcina and A. Silvestri, "Validation and application of a reliability allocation technique (advanced integrated factors method) to an industrial system," in *Proceedings of the International Conference on Modelling, Identification and Control*, 2014, pp. 75–79.

[45] H. Biedermann, "Instandhaltung und Wissensmanagement: Die Entwicklung zur lernenden Instandhaltungsorganisation," *Industrie-Management (2002)*, vol. 2, pp. 9–12, 2002.

[46] S. Bosse, D. Lehmhus, W. Lang and M. Busse, *Material-Integrated Intelligent Systems: Technology and Applications*. Wiley, 2018.

[47] R. K. Mobley, *Maintenance Fundamentals*. Butterworth Heinemann, 2011, vol. 2.

[48] H. Mikat, "Hybride Fehlerprognose zur Unterstützung prädiktiver Instandhaltungskonzepte in der Luftfahrt," Ph.D. dissertation, Technische Universität, 2015.

[49] Deutsches Institut für Normung e. V., *Grundlagen der Instandhaltung*, DIN 31051: 2019-06 (DIN), 2019.

[50] S. Kadry, *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*. Engineering Science Reference, 2012.

[51] F. Ansari, R. Glawar and T. Nemeth, "PriMa: a prescriptive maintenance model for cyber-physical production systems," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 4-5, pp. 482–503, 2019.

[52] K. Matyas, T. Nemeth, K. Kovacs and R. Glawar, "A procedural approach for realizing prescriptive maintenance planning in manufacturing industries," *CIRP Annals*, vol. 66, no. 1, pp. 461–464, 2017.

[53] Andrew K.S. Jardine, Daming Lin and Dragan Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[54] A. Cachada, J. Barbosa, P. Leitño, *et al.*, "Maintenance 4.0: intelligent and predictive maintenance system architecture," in *Proceedings of the 23rd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2018, pp. 139–146.

[55] T. Produ, L. Laperrière and G. Reinhart, Eds., *CIRP Encyclopedia of Production Engineering*. Springer, 2014.

[56] B. S. Dhillon, *Engineering Maintenance: A Modern Approach*. CRC Press, 2002.

[57] A. Birolini, *Reliability Engineering: Theory and Practice*, ser. Engineering online library. Springer, 2007, vol. 5.

[58] G. J. Vachtsevanos, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Wiley, 2006.

[59] K. Efthymiou, N. Papakostas, D. Mourtzis and G. Chryssolouris, "On a predictive maintenance platform for production systems," *Procedia CIRP*, vol. 3, pp. 221–226, 2012.

[60] R. Gouriveau, K. Medjaher and N. Zerhouni, *From Prognostics and Health Systems Management to Predictive Maintenance 1: Monitoring and Prognostics*. Wiley, 2016, vol. 4.

[61] I. Heinrich, *Ausserplaninstandhaltung von Eisenbahnfahrzeugen: Algorithmus für einen vorhersagbaren Schadindex*. Springer, 2017.

[62] M. Strunz, *Instandhaltung: Grundlagen-Strategien-Werkstätten*. Springer-Verlag, 2012.

[63] T. Tinga and R. Loendersloot, "Physical model-based prognostics and health monitoring to enable predictive maintenance," in *Predictive Maintenance in Dynamic Systems*, Springer, 2019, pp. 313–353.

[64] Deutsches Institut für Normung e. V., *Condition monitoring and diagnostics of machines - data processing, communication and presentation*, DIN ISO 13374-1: 2003 (DIN ISO), 2003.

[65] Antonio J. Guillén, Adolfo Crespo, Juan Fco. Gómez and Maria Dolores Sanz, "A framework for effective management of condition based maintenance programs in the context of industrial development of E-Maintenance strategies," *Computers in Industry*, vol. 82, pp. 170–185, 2016.

[66] A. C. Márquez, V. G.-P. Díaz and J. F. G. Fernández, *Advanced Maintenance Modelling for Asset Management: Techniques and Methods for Complex Industrial Systems*. 2017.

[67] International Organization for Standardization, *Condition monitoring and diagnostics of machines - general guidelines*, ISO 17359: 2018(E), 2018.

[68] G. Dong and H. Liu, *Feature Engineering for Machine Learning and Data Analytics*. CRC Press, 2018.

[69] M. Lebold and M. Thurston, "Open standards for condition-based maintenance and prognostic systems," in *Proceedings of the Maintenance and Reliability Conference (MARCON)*, vol. 200, 2001.

[70] K. Holmberg, A. Adgar, A. Arnaiz, E. Jantunen, J. Mascolo and S. Mekid, *E-maintenance*. Springer, 2010.

[71] A. Mathew, L. Zhang, S. Zhang and L. Ma, "A review of the MIMOSA OSA-EAI database for condition monitoring systems," in *Engineering Asset Management*, Springer, 2006, pp. 837–846.

[72] G. Niu, B.-S. Yang and M. Pecht, "Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance," *Reliability Engineering & System Safety*, vol. 95, no. 7, pp. 786–796, 2010.

[73] H. O. Ahmed and A. K. Nandi, "Intelligent condition monitoring for rotating machinery using compressively-aampled data and sub-space learning techniques," in *Proceedings of the 10th International Conference on Rotor Dynamics (IFToMM)*, 2018, pp. 238–251.

[74] H. Han, B. Gu, J. Kang and Z. R. Li, "Study on a hybrid SVM model for chiller FDD applications," *Applied Thermal Engineering*, vol. 31, no. 4, pp. 582–592, 2011.

[75] C. Laughman, K. Lee, R. Cox, *et al.*, "Power signature analysis," *IEEE Power and Energy Magazine*, vol. 1, no. 2, pp. 56–63, 2003.

[76] J. Navarro-Esbri, E. Torrella and R. Cabello, "A vapour compression chiller fault detection technique based on adaptative algorithms. Application to on-line refrigerant leakage detection," *International Journal of Refrigeration*, vol. 29, no. 5, pp. 716–723, 2006.

[77] R. Islamaj, L. Getoor and W. J. Wilbur, "A feature generation algorithm for sequences with application to splice-site prediction," in *Proceedings of the 10th European Conference on Principles of Data Mining and Knowledge Discovery*, 2006, pp. 553–560.

[78] R. L. Hu, "Machine Learning to Scale Fault Detection in Smart Energy Generation and Building Systems," Ph.D. dissertation, UC Berkeley, 2016.

[79] K. F. Draiss, *Advanced Man-Machine Interaction, Fundamentals and Implementation. Signals and Communication Technology*, 2006.

[80] P. K. Pattnaik, S. Mohanty and S. Mohanty, *Smart Healthcare Analytics in IoT Enabled Environment*. Springer, 2020.

[81] E. L. Russell, L. H. Chiang and R. D. Braatz, *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*. Springer, 2012.

[82] A. C. Olivieri, G. M. Escandar, H. C. Goicoechea and A. de la Peña, *Fundamentals and Analytical Applications of Multiway Calibration*. Elsevier, 2015.

[83] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of Bioinformatics and Computational Biology*, vol. 3, no. 02, pp. 185–205, 2005.

[84] A. Berge and A. S. Solberg, "Improving hyperspectral classifiers: the difference between reducing data dimensionality and reducing classifier parameter complexity," in *Scandinavian Conference on Image Analysis*, 2007, pp. 293–302.

[85] K. Yan, Z. Ji and W. Shen, "Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM," *Neurocomputing*, vol. 228, pp. 205–212, 2017.

[86] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. CRC Press, 2007.

[87] G. Li, Y. Hu, H. Chen, *et al.*, "An improved fault detection method for incipient centrifugal chiller faults using the PCA-R-SVDD algorithm," *Energy and Buildings*, vol. 116, pp. 104–113, 2016.

[88] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki and J. Schröder, *Diagnosis and Fault-Tolerant Control*. Springer, 2006, vol. 2.

[89] V. Palade and C. D. Bocaniala, *Computational Intelligence in Fault Diagnosis*. Springer, 2006.

[90] F. Jiang, W. Li, Z. Wang and Z. Zhu, "Fault severity estimation of rotating machinery based on residual signals," *Advances in Mechanical Engineering*, vol. 4, 2012.

[91] A. Beghi, L. Cecchinato, C. Corazzol, M. Rampazzo, F. Simmini and G. A. Susto, "A one-class SVM based tool for machine learning novelty detection in HVAC chiller systems," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1953–1958, 2014.

[92] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, 2008.

[93] A. Alzghoul, B. Backe, M. Löfstrand, A. Byström and B. Liljedahl, "Comparing a knowledge-based and a data-driven method in querying data streams for system fault detection: A hydraulic drive system application," *Computers in Industry*, vol. 65, no. 8, pp. 1126–1135, 2014.

[94] R. Arunthavanathan, F. Khan, S. Ahmed and S. Imtiaz, "An analysis of process fault diagnosis methods from safety perspectives," *Computers & Chemical Engineering*, 2020.

[95] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," in *IEEE Transactions on Industrial Informatics*, vol. 9, 2013, pp. 2226–2238.

[96] D. Jung, K. Y. Ng, E. Frisk and M. Krysander, "Combining model-based diagnosis and data-driven anomaly classifiers for fault isolation," *Control Engineering Practice*, vol. 80, pp. 146–156, 2018.

[97] S. X. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer, 2008.

[98] Y. Zhao, S. Wang and F. Xiao, "A statistical fault detection and diagnosis method for centrifugal chillers based on exponentially-weighted moving average control charts and support vector regression," *Applied Thermal Engineering*, vol. 51, no. 1-2, pp. 560–572, 2013.

[99] D. A. T. Tran, Y. Chen, H. L. Ao and H. N. T. Cam, "An enhanced chiller FDD strategy based on the combination of the LSSVR-DE model and EWMA control charts," *International Journal of Refrigeration*, vol. 72, pp. 81–96, 2016.

[100] F. Xiao, C. Zheng and S. Wang, "A fault detection and diagnosis strategy with enhanced sensitivity for centrifugal chillers," *Applied Thermal Engineering*, vol. 31, no. 17-18, pp. 3963–3970, 2011.

[101] Y. Zhao, F. Xiao and S. Wang, "An intelligent chiller fault detection and diagnosis methodology using Bayesian belief network," *Energy and Buildings*, vol. 57, pp. 278–288, 2013.

[102] A. Beghi, L. Cecchinato, F. Peterle, M. Rampazzo and F. Simmini, "Model-based fault detection and diagnosis for centrifugal chillers," in *Proceedings of the 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, 2016, pp. 158–163.

[103] L. H. Chiang, E. L. Russell and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer, 2000.

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

[105] S. Estrada-Flores, I. Merts, B. de Ketelaere and J. Lammertyn, "Development and validation of "grey-box" models for refrigeration applications: A review of key concepts," *International Journal of Refrigeration*, vol. 29, no. 6, pp. 931–946, 2006.

[106] Y. Nakkabi, N. Kabbaj, B. Dahhou, G. Roux and J. Aguilar-Martin, "A combined analytical and knowledge based method for fault detection and isolation," in *Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 2, 2003, pp. 161–164.

[107] J. Schein and S. T. Bushby, "A hierarchical rule-based fault detection and diagnostic method for HVAC systems," *HVAC&R Research*, vol. 12, no. 1, pp. 111–125, 2006.

[108] C. Nan, F. Khan and M. T. Iqbal, "Real-time fault diagnosis using knowledge-based expert system," *Process Safety and Environmental Protection*, vol. 86, no. 1, pp. 55–71, 2008.

[109] C.-M. Fan and Y.-P. Lu, "A Bayesian framework to integrate knowledge-based and data-driven inference tools for reliable yield diagnoses," in *Proceedings of the Winter Simulation Conference 2008*, 2008, pp. 2323–2329.

[110] S. M. El-Shal and A. S. Morris, "A fuzzy expert system for fault detection in statistical process control of industrial processes," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 2, pp. 281–289, 2000.

[111] J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[112] V. Chandola, A. Banerjee and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[113] K. Yan, C. Zhong, Z. Ji and J. Huang, "Semi-supervised learning for early detection and diagnosis of various air handling unit faults," *Energy and Buildings*, vol. 181, pp. 75–83, 2018.

[114] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor and J. C. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems*, vol. 12, MIT Press, 2000, pp. 582–588.

[115] C. C. Aggarwal, *Outlier Analysis*. Springer New York, 2013.

[116] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[117] Hua Han, Ling Xu, Xiaoyu Cui and Yuqiang Fan, "Novel chiller fault diagnosis using deep neural network (DNN) with simulated annealing (SA)," *International Journal of Refrigeration*, vol. 121, pp. 269–278, 2021.

[118] R. Shrivastava, H. Mahalingam and N. N. Dutta, "Application and evaluation of random forest classifier technique for fault detection in bioreactor operation," *Chemical Engineering Communications*, vol. 204, no. 5, pp. 591–598, 2017.

[119] T. Yairi, Y. Kato and K. Hori, "Fault detection by mining association rules from house-keeping data," in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, vol. 18, 2001, p. 21.

[120] F. Harrou, A. Dairi, B. Taghezouit and Y. Sun, "An unsupervised monitoring procedure for detecting anomalies in photovoltaic systems using a one-class Support Vector Machine," *Solar Energy*, vol. 179, pp. 48–58, 2019.

[121] D. Li, G. Hu and C. J. Spanos, "A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis," *Energy and Buildings*, vol. 128, pp. 519–529, 2016.

[122] L. Liao and F. Köttig, "A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction," *Applied Soft Computing*, vol. 44, pp. 191–199, 2016.

[123] S. Frank, M. Heaney, X. Jin, *et al.*, "Hybrid model-based and data-driven fault detection and diagnostics for commercial buildings," in *Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings*, 2016.

[124] G. Jark-Heinrich and E. K. Antonsson, *Springer Handbook of Mechanical Engineering*. Springer, 2009, vol. 10.

[125] J. Dohmann, *Thermodynamik der Kälteanlagen und Wärmepumpen*. Springer, 2016.

[126] D. V. Schroeder, *An Introduction to Thermal Physics*. Oxford University Press, 2021.

[127] H. W. Stanford, *HVAC Water Chillers and Cooling Towers: Fundamentals, Application, and Operation*, ser. Mechanical Engineering. CRC Press, 2003.

[128] Z. Du, P. A. Domanski and W. V. Payne, "Effect of common faults on the performance of different types of vapor compression systems," *Applied Thermal Engineering*, vol. 98, pp. 61–72, 2016.

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

[130] M. C. Comstock, J. E. Braun and E. A. Groll, "A survey of common faults for chillers," *Ashrae Transactions*, vol. 108, 2002.

[131] J. Cui and S. Wang, "A model-based online fault detection and diagnosis strategy for centrifugal chiller systems," *International Journal of Thermal Sciences*, vol. 44, no. 10, pp. 986–999, 2005.

[132] H. Han, X. Cui, Y. Fan and H. Qing, "Least squares support vector machine (LS-SVM)-based chiller fault diagnosis using fault indicative features," *Applied Thermal Engineering*, vol. 154, pp. 540–547, 2019.

[133] M. C. Comstock, J. E. Braun and R. Bernhard, *Development of Analysis Tools for the Evaluation of Fault Detection and Diagnostics in Chillers*. Purdue University, 1999.

[134] Z. Wang, Z. Wang, S. He, X. Gu and Z. F. Yan, "Fault detection and diagnosis of chillers using Bayesian network merged distance rejection and multi-source non-sensor information," *Applied Energy*, vol. 188, pp. 200–214, 2017.

[135] D. A. T. Tran, Y. Chen, M. Q. Chau and B. Ning, "A robust online fault detection and diagnosis strategy of centrifugal chiller systems for building energy efficiency," *Energy and Buildings*, vol. 108, pp. 441–453, 2015.

[136] D. A. T. Tran, Y. Chen and C. Jiang, "Comparative investigations on reference models for fault detection and diagnosis in centrifugal chiller systems," *Energy and Buildings*, vol. 133, pp. 246–256, 2016.

[137] S. He, Z. Wang, Z. Wang, X. Gu and Z. Yan, "Fault detection and diagnosis of chiller using Bayesian network classifier with probabilistic boundary," *Applied Thermal Engineering*, vol. 107, pp. 37–47, 2016.

[138] Y. Zhao, S. Wang and F. Xiao, "Pattern recognition-based chillers fault detection method using support vector data description (SVDD)," *Applied Energy*, vol. 112, pp. 1041–1048, 2013.

[139] H. Han, Z. Cao, B. Gu and N. Ren, "PCA-SVM-based automated fault detection and diagnosis (AFDD) for vapor-compression refrigeration systems," *HVAC&R Research*, vol. 16, no. 3, pp. 295–313, 2010.

[140] Q. Zhou, S. Wang and F. Xiao, "A novel strategy for the fault detection and diagnosis of centrifugal chiller systems," *HVAC&R Research*, vol. 15, no. 1, pp. 57–75, 2009.

[141] B. Schölkopf, A. J. Smola, F. Bach, *et al.*, *Learning with Kernels: Support Vector machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[142] Y. Zhao, F. Xiao, J. Wen, Y. Lu and S. Wang, "A robust pattern recognition-based fault detection and diagnosis (FDD) method for chillers," *HVAC&R Research*, vol. 20, no. 7, pp. 798–809, 2014.

[143] K. Yan and J. Hua, "Deep learning technology for chiller faults diagnosis," in *2019 IEEE International Conferance on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress*, 2019, pp. 72–79.

[144] Z. Zhang, H. Han, X. Cui and Y. Fan, "Novel application of multi-model ensemble learning for fault diagnosis in refrigeration systems," *Applied Thermal Engineering*, vol. 164, 2020.

[145] K. Yan, W. Shen, T. Mulumba and A. Afshari, "ARX model based fault detection and diagnosis for chillers using support vector machines," *Energy and Buildings*, vol. 81, pp. 287–295, 2014.

[146] G. Csurka, "A comprehensive survey on domain adaptation for visual applications," in *Domain Adaptation in Computer Vision Applications*, G. Csurka, Ed., Springer, 2017, pp. 1–35.

[147] B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2007.

[148] H. Zheng, R. Wang, Y. Yang, *et al.*, "Cross-domain fault diagnosis using knowledge transfer strategy: a review," in *IEEE Access*, vol. 7, 2019, pp. 129 260–129 290.

[149] H. Han, B. Gu, T. Wang and Z. R. Li, "Important sensors for chiller fault detection and diagnosis (FDD) from the perspective of feature selection and machine learning," *International Journal of Refrigeration*, vol. 34, no. 2, pp. 586–599, 2011.

[150] A. S. Glass, P. Gruber, M. Roos and J. Todtli, "Qualitative model-based fault detection in air-handling units," *IEEE Control Systems Magazine*, vol. 15, no. 4, pp. 11–22, 1995.

[151] X. Xu, F. Xiao and S. Wang, "Enhanced chiller sensor fault detection, diagnosis and estimation using wavelet analysis and principal component analysis methods," *Applied Thermal Engineering*, vol. 28, no. 2-3, pp. 226–237, 2008.

[152] S. Wang and J. Cui, "A robust fault detection and diagnosis strategy for centrifugal chillers," *HVAC&R Research*, vol. 12, no. 3, pp. 407–428, 2006.

[153] C. C. Aggarwal, *Data Mining: The Textbook*. Springer, 2015.

[154] G. Li, Y. Hun, H. Chen and H. Li, "Study on the support vector data description (SVDD)-based chiller sensor fault detection efficiencies," in *The 24th IIR International Congress of Refrigeration, Yokohama, Japan*, 2015.

[155] S. S. Khan and M. G. Madden, "A survey of recent trends in one class classification," in *Irish Conference on Artificial Intelligence and Cognitive Science*, 2009, pp. 188–197.

[156] R. Kiryo, G. Niu, M. C. Du Plessis and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," *Computing Research Repository (CoRR)*, 2017.

[157] B. Liu, Y. Dai, X. Li, W. S. Lee and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in *Proceedings of the 3rd IEEE International Conference on Data Mining*, 2003, pp. 179–186.

[158] S. Valle, W. Li and S. J. Qin, "Selection of the number of principal components: the variance of the reconstruction error criterion with a comparison to other methods," *Industrial & Engineering Chemistry Research*, vol. 38, no. 11, pp. 4389–4401, 1999.

[159] J. Cui and S. Wang, "A model-based online fault detection and diagnosis strategy for centrifugal chiller systems," *International Journal of Thermal Sciences*, vol. 44, no. 10, pp. 986–999, 2005.

[160] K. Hempstalk, E. Frank and I. H. Witten, "One-class classification by combining density and class probability estimation," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2008, pp. 505–519.

[161] N. Cristianini, J. Shawe-Taylor, *et al.*, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[162] V. Vapnik, S. E. Golowich, A. Smola, *et al.*, "Support vector method for function approximation, regression estimation, and signal processing," *Advances in Neural Information Processing Systems*, pp. 281–287, 1997.

[163] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 111–133, 1974.

[164] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[165] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 3, 2003, pp. 587–592.

[166] J. Mũnoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone and G. Camp-Valls, "Semisupervised one-class support vector machines for classification of remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 8, pp. 3188–3197, 2010.

[167] H. Wang, L. Zhang, Y. Xiao and W. Xu, "An approach to choosing gaussian kernel parameter for one-class SVMs via tightness detecting," in *Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, 2012, pp. 318–323.

[168] Q. Yang, Y. Zhang, W. Dai and S. J. Pan, *Transfer Learning*. Cambridge University Press, 2020.

[169] Z. Chai and C. Zhao, "A fine-grained adversarial network method for cross-domain industrial fault diagnosis," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1432–1442, 2020.

[170] H. Zhao, R. T. D. Combes, K. Zhang and G. Gordon, "On learning invariant representations for domain adaptation," in *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, 2019, pp. 7523–7532.

[171] X. Li, W. Zhang, Q. Ding and J.-Q. Sun, "Multi-layer domain adaptation method for rolling bearing fault diagnosis," *Signal Processing*, vol. 157, pp. 180–197, 2019.

[172] A. Margolis, "A literature review of domain adaptation with unlabeled data," *Technical Report*, pp. 1–42, 2011.

[173] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf and A. Smola, "A kernel method for the two-sample-problem," *Advances in Neural Information Processing Systems*, vol. 19, pp. 513–520, 2006.

[174] B. Yang, Y. Lei, F. Jia and S. Xing, "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings," *Mechanical Systems and Signal Processing*, vol. 122, pp. 692–706, 2019.

[175] Q. Wang, G. Michau and O. Fink, "Domain adaptive transfer learning for fault diagnosis," in *Prognostics and System Health Management Conference (PHM-Paris)*, 2019, pp. 279–285.

[176] Y. Ganin, E. Ustinova, H. Ajakan, *et al.*, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1–35, 2016.

[177] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2014.

[178] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 325–333.

[179] Y. Fan, S. Nowaczyk and T. Rögnvaldsson, "Transfer learning for remaining useful life prediction based on consensus self-organizing models," *Reliability Engineering & System Safety*, vol. 203, p. 107 098, 2020.

[180] S. J. Pan, I. W. Tsang, J. T. Kwok and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[181] B. Sun, J. Feng and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[182] B. Fernando, A. Habrard, M. Sebban and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2960–2967.

[183] B. Kulis, K. Saenko and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1785–1792.

[184] M. Harel and S. Mannor, "Learning from multiple outlooks," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2010.

[185] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[186] C.-W. Hsu, C.-C. Chang and C.-J. Lin, "A practical guide to support vector classification," *Technical Report*, 2003.

[187] R. Debnath, N. Takahide and H. Takahashi, "A decision based one-against-one method for multi-class support vector machine," *Pattern Analysis and Applications*, vol. 7, no. 2, pp. 164–175, 2004.

[188] J. C. Platt, N. Cristianini, J. Shawe-Taylor, *et al.*, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems: Proceedings of the 1999 Conference*, vol. 12, 1999, pp. 547–553.

[189] M. Blesl and A. Kessler, "Gekoppelte und sonstige Querschnittsprozesse," in *Energieeffizienz in der Industrie*, Berlin, Heidelberg: Springer, 2017, pp. 239–286.

[190] M. Kind and H. Martin, "VDI-Wärmeatlas," 2013.

[191] T. Gao, M. David, J. Geer, R. Schmidt and B. Sammakia, "Experimental and numerical dynamic investigation of an energy efficient liquid cooled chiller-less data center test facility," *Energy and Buildings*, vol. 91, pp. 83–96, 2015.

[192] P. R. Armstrong, C. R. Laughman, S. B. Leeb and L. K. Norford, "Detection of rooftop cooling unit faults based on electrical measurements," *HCAV&R Research*, vol. 12, no. 1, pp. 151–175, 2006.

[193] M. Kim, S. H. Yoon, P. A. Domanski and W. V. Payne, "Design of a steady-state detector for fault detection and diagnosis of a residential air conditioner," *International Journal of Refrigeration*, vol. 31, no. 5, pp. 790–799, 2008.

[194] T. Duncan, "The rotary screw compressor," *ASHRAE Journal*, vol. 41, no. 5, p. 34, 1999.

[195] B. E. Poling, J. M. Prausnitz and J. P. O'connell, *Properties of Gases and Liquids*. McGraw-Hill Education, 2001.

[196] B. Sundén and R. M. Manglik, *Plate Heat Exchangers: Design, Applications and Performance*. WIT Press, 2007, vol. 11.

[197] A. Reinhart, "Kälteversorgung," in *Raumklimatechnik–Band 2*, Springer, 2008, pp. 439–519.

[198] M. Borg, C. Englund, K. Wnuk, *et al.*, "Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry," *Journal of Automotive Software Engineering*, vol. 1, no. 1, pp. 1–19, 2020.

[199] D. J. Hand and S. Khan, "Validating and verifying AI systems," *Patterns*, vol. 1, no. 3, 2020.

[200] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, 2016.

[201] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[202] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[203] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, 2020.

[204] M. Andersen, J. Dahl, Z. Liu, *et al.*, "Interior-point methods for large-scale cone programming," *Optimization for Machine Learning*, vol. 54, 2011.

[205] I. Kononenko, E. Šimec and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with RELIEFF," *Applied Intelligence*, vol. 7, no. 1, pp. 39–55, 1997.

[206] M. Sokolova, N. Japkowicz and S. Szpakowicz, "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation," in *Proceedings of the Australasian Joint Conference on Artificial Intelligence*, 2006, pp. 1015–1021.

[207] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020.

[208] V. van Asch, "Macro-and micro-averaged evaluation measures," *Computational Linguistics & Psycholinguistics*, vol. 49, 2013.

[209] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[210] P. A. Flach, "The geometry of ROC space: understanding machine learning metrics through ROC isometrics," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 194–201.

[211] C. F. Alcala and S. J. Qin, "Reconstruction-based contribution for process monitoring," *Automatica*, vol. 45, no. 7, pp. 1593–1600, 2009.

A Appendix

This chapter completes this dissertation with important information on the overall CBM model development. This information includes the basics for the algorithm implementations shown in pseudo-code, which might be especially important for practitioners. Furthermore, additional information regarding the data collection phase are provided, addressing both the experimental procedure as well as additional results, or fault patterns to be precise. Lastly, the classification performance obtained from the proposed model is visualised.

A.1 Algorithm Implementation

This section specifically addresses the algorithm implementation concerning both the model training but also the deployment phase.

A.1.1 Steady-State Detector

Algorithm A.1 shows the pseudo-code for the implementation of the steady-state detector.

Algorithm A.1 The steady-state detector.

Input
  X_{raw} Unfiltered raw data arranged according to the time of measurement
  $sfeat$ Collection of steady-sate indicating features ▷ Extracted from X_{raw}
  pd Polynomial degree of Savitzky-Golay filter ▷ In this work $\rightarrow pd = 3$
  δ_t The pre-set scalar defining the steady-state threshold
  ws The window size in which polynomial approximation is performed
  t_{min} The minimum steady-state time
Output
  X_{steady} The matrix containing only steady-state data
$dev \leftarrow$ Compute time derivatives from $sfeat$
$\sigma_{dev} \leftarrow$ Compute standard deviation for each feature time derivative in dev
$dev \leftarrow savitzky\_golay\_filter(dev, pd, ws)$ ▷ Filter the time derivatives
$X_{raw} \leftarrow$ Initialise empty steady-state matrix
$start\ time \leftarrow$ Initialise time variable with first $timestamp$ of dev
for $\langle timestamp, i \rangle$ in dev **do** ▷ Loop through all indices and timestamps of dev
  for $feat$ in dev **do** ▷ Loop through all steady-state features
    if $(dev[i, feat] < \delta_t \cdot \sigma_{dev}/2)$ and $(dev[i, feat] > -\delta_t \cdot \sigma_{dev}/2)$ **then**
      if $timestamp - start\ time \geq t_{min}$ **then**
        add observation $X_{raw}[i, :]$ to X_{steady}
      end if
    else
      $start\ time = timestamp$
    end if
  end for
end for
return X_{steady}

A.1.2 CBM Model Training

The CBM model training process is demonstrated in Algorithm A.2. Note that the design matrices of the source and target domain contain filtered and steady-state data.

Algorithm A.2 The implementation of the CBM model training process.

Input
- X_s Source domain design matrix with k features and $n_s - n_{ns}$ samples
- Y_s Source domain labels containing m fault classes and $n_s - n_{ns}$ labels
- X_t Target domain design matrix with k features and n_t samples
- Y_t Target domain labels $Y_t \in \{-1, 1\}$ (unlabelled/normal), with n_t labels
- $CPV_{k_{pc}}$ Minimum cumulative percent variance to be discarded

Output
- k_{pc} The number of principal components to be discarded
- pca_{fd} The trained target domain PCA algorithm used for fault detection
- pca_{fi} The trained target domain PCA algorithm used for fault isolation
- $h(\cdot)_{fd}$ The fault detection model
- $h(\cdot)_{fi}$ The fault isolation model

$Z \leftarrow$ Select only fault-free data instances from X_t where $y_{t_i} = 1$
$pca_{fd} \leftarrow train(\text{PCA}, Z)$ ▷ Initialise PCA for fault detection
$CPV = 0$
for $pcindex = 1 < k$ **do** ▷ Loop through the principal component indices
  $CPV = CPV + get\_explained\_variance(pca_{fd}, pcindex)$ ▷ Variance for each PC
  if $CPV > CPV_{k_{pc}}$ **then**
    $k_{pc} \leftarrow pcindex$
    break ▷ Exit the for loop and continue
  end if
end for
$\tilde{X}_t \leftarrow transform\_to\_rcs(X_t, pca_{fd}, k_{pc})$
$bsvm\ grid \leftarrow$ define BSVM search space ▷ The hyperparameter search space
$h(\cdot)_{bsvm} \leftarrow train(h(\cdot)_{bsvm}, bsvm\ grid, \tilde{X}_t, Y_t)$
for $i = 0 < n_t$ **do** ▷ Update target domain training labels
  if $Y_t[i] = -1$ **then**
    if $h(\tilde{X}_t[i, :])_{bsvm} = 1$ **then** $Y_t[i] = 1$
    end if
  end if
end for
$h(\cdot)_{ocsvm} \leftarrow train(h(\cdot)_{ocsvm}, \tilde{X}_t, Y_t)$ ▷ Optimise OC-SVM via *tightness detection*
$h(\cdot)_{fd} \leftarrow \langle h(\cdot)_{bsvm}, h(\cdot)_{ocsvm} \rangle$ ▷ Stack BSVM and OC-SVM
$Z_{nt} \leftarrow$ Select only fault-free data instances from X_t where $y_{t_i} = 1$
$pca_{fi} \leftarrow train(\text{PCA}, Z_{nt})$ ▷ Train PCA with updated target domain labels
$Z_{fs} \leftarrow$ Select only fault data from X_s
$Y_{fs} \leftarrow$ Select only fault labels from Y_s
$\tilde{Z}_{fs} \leftarrow transform\_to\_rcs(Z_{fs}, pca_{fi}, k_{pc})$
$svm\ grid \leftarrow$ define SVM search space
$h(\cdot)_{fi} \leftarrow train(h(\cdot), svm\ grid, \tilde{Z}_{fs}, Y_{fs})$ ▷ Collection of $(m-1)m/2$ bin. classifiers
return $pca_{fd}, pca_{fi}, h(\cdot)_{fd}, h(\cdot)_{fi}$

A.1.3 CBM Model Deployment

Algorithm A.3 shows the pseudo-code of the CBM model deployment based on any unknown observation x_t^u to be evaluated. The outcome is the fault type index (fti) and the associated health index (hi). Note that the latter is chosen to be the smaller value of the BSVM or the OC-SVM algorithm.

Algorithm A.3 The implementation of the CBM model deployment.

Input
  x_t^u Unknown observations
  k_{pc} The number of principal components to be discarded
  pca_{fd} The trained target domain PCA algorithm used for fault detection
  pca_{fi} The trained target domain PCA algorithm used for fault isolation
  $h(\cdot)_{fd}$ The fault detection model
  $h(\cdot)_{fi}$ The fault isolation model
Output
  fti The fault type index
  hi The health index
$\tilde{x}_t^u \leftarrow transform\_to\_rcs(x_t^u, pca_{fd}, k_{pc})$
if $h(\tilde{x}_t^u)_{fd} = 1$ **then** ▷ Perform fault detection
  $fti = None$ ▷ No fault index to be indicated
  $hi = 10$ ▷ Fully functioning system
else ▷ Novelty detected
  $\tilde{z}_t^u \leftarrow transform\_to\_rcs(x_t^u, pca_{fi}, k_{pc})$ ▷ Using PCA for fault isolation
  $votes \leftarrow$ Initialise empty voting array
  for $i = 0 < m$ **do** ▷ Loop through fault class indices
    for $j = 0 < m$ **do** ▷ Loop through each combination $h(\cdot)_{i,j}$ in $h(\cdot)_{fi}$
      if $i \neq j$ **then**
        $votes[i] = votes[i] + f(h(\tilde{z}_t^u)_{i,j})$ ▷ Get the vote for or against class i
      end if
    end for
  end for
  $fti = argmax(votes)$
  if $length(fti) > 1$ **then** ▷ Multiple classes received the same number of votes
    $fti \leftarrow$ Choose fault type index with largest outcome of decision function $h(\cdot)^*_{i,j}$
  end if
  $h(\cdot)^*_{bsvm}, h(\cdot)^*_{ocsvm} \leftarrow$ get decision functions from $h(\cdot)_{fd}$
  $hi_{bsvm} = scale\_to\_hi(h(\tilde{x}_t^u)^*_{bsvm})$ ▷ Scale outcome to health index range
  $hi_{ocsvm} = scale\_to\_hi(h(\tilde{x}_t^u)^*_{ocsvm})$
  $hi = min([hi_{bsvm}, hi_{ocsvm}])$ ▷ Get the minimum health index
end if
return fti, hi

A.2 Test Sequence of Set-Points

Table A.1 shows the test sequence applied throughout the data acquisition phase. Three control variables were used to derive the sequence consisting out of 27 operational states, namely: (1) the temperature of the water-glycol entering the evaporator (T^*_{ei}), (2) the temperature of the water-glycol entering the condenser (T^*_{ci}) and (3) the compressor rotational speed (n^*_{comp}).

Table A.1: The test sequence of the data acquisition phase.

Test	T^*_{ei} [°C]	T^*_{ci} [°C]	n^*_{comp} [%] ([min^{-1}])
1	5	23	65 (2930)
2	5	23	50 (2254)
3	5	23	35 (1578)
4	10	23	65 (2930)
5	10	23	50 (2254)
6	10	23	35 (1578)
7	15	23	65 (2930)
8	15	23	50 (2254)
9	15	23	35 (1578)
10	5	28	65 (2930)
11	5	28	50 (2254)
12	5	28	35 (1578)
13	10	28	65 (2930)
14	10	28	50 (2254)
15	10	28	35 (1578)
16	15	28	65 (2930)
17	15	28	50 (2254)
18	15	28	35 (1578)
19	5	33	65 (2930)
20	5	33	50 (2254)
21	5	33	35 (1578)
22	10	33	65 (2930)
23	10	33	50 (2254)
24	10	33	35 (1578)
25	15	33	65 (2930)
26	15	33	50 (2254)
27	15	33	35 (1578)

A.3 Sequence of Conducted Experiments

Table A.2 contains the experiments conducted in this thesis in chronological order, starting with the commissioning phase and ending with the fault test procedure. The experiments designated with 'Cancelled' refer to the experiments after a fault has been attempted to be removed but failed to restore the original operating condition. On the other hand, both 'RL0-1', 'RL0-2', etc., refer to experiments with induced faults not impacting the overall chiller operating condition (see Section 4.4.5 for more details).

Table A.2: Overview of all conducted experiments during the data acquisition phase.

Date	Acronym	Used	Description
17-12-2018	CMG1	No	First commissioning - PID controller optimisation
18-12-2018	CMG2	No	First implementation of the test sequence
19-12-2018	CMG3	No	Test sequence adapted to high pressure emergency shut-down
20-12-2018	CMG4	No	Repeated test sequence
03-01-2019	CMG5	No	Tests after removing retained refrigerant oil
04-01-2019	CMG6	No	Repeated test sequence
07-01-2019	CMG7	No	Tests after removing software errors
08-03-2019	BM	Yes	Benchmark test
11-03-2019	RVE1	Yes	SL1-15 % reduced evaporator volume flow rate
12-03-2019	RVE2	Yes	SL2-30 % reduced evaporator volume flow rate
13-03-2019	RVE3	Yes	SL3-45 % reduced evaporator volume flow rate
14-03-2019	RVE4	Yes	SL4-60 % reduced evaporator volume flow rate
15-03-2019	N-RVE	Yes	Normal test
19-03-2019	RVC1	Yes	SL1-7.5 % reduced condenser volume flow rate
22-03-2019	RVC2	Yes	SL2-15 % reduced condenser volume flow rate
27-03-2019	RVC3	Yes	SL3-22.5 % reduced condenser volume flow rate
02-04-2019	RVC4	Yes	SL4-30 % reduced condenser volume flow rate
04-04-2019	N-RVC	Yes	Normal test
25-04-2019	NC1	Yes	SL1-10 % nitrogen added to refrigerant line
26-04-2019	NC2	Yes	SL2-20 % nitrogen added to refrigerant line
07-05-2019	NC3	Yes	SL3-30 % nitrogen added to refrigerant line
08-05-2019	NC4	Yes	SL4-40 % nitrogen added to refrigerant line
28-05-2019	N-NC-Cancelled1	No	Normal test- normal operation condition could not be restored (nitrogen was still present within the system)
13-06-2019	N-NC-Cancelled2	No	Normal test- normal operation condition could not be restored (nitrogen was still present within the system)
15-08-2019	N-NC	Yes	Normal test- nitrogen successfully removed
26-08-2019	RL0-1	No	SL0-10 % refrigerant removed
27-08-2019	RL0-2	No	SL0-20 % refrigerant removed
28-08-2019	RL0-3	No	SL0-30 % refrigerant removed
29-08-2019	RL0-4	No	SL0-40 % refrigerant removed
10-09-2019	RL1	Yes	SL1-50 % refrigerant removed
12-09-2019	RL2	Yes	SL2-60 % refrigerant removed

A.4 Results from the Fault Simulation Phase

Following from Section 4.4, this section extends the overview of the discussed fault patterns based on the previously introduced regression analysis. Thereby, the plots shown below illustrate the most important changes in the operating condition of the screw-chiller being part of the test rig in the presence of certain faults. To emphasise the impact of the the respective fault severity, the figures show the deviations for the four severity levels determined for the experimental procedure in this work.

It is worth noting that for reasons of for simplified representations, the plots show the fault patterns for the set-point $T_{ei}^{*} = 10$ °C. For the average deviation from the benchmark dataset across all operating conditions, the reader is referred to the tables provided in Section 4.4.

A.4.1 Reduced Evaporator Water-Flow Rate

The fault RVE was primary accompanied by a pressure drop if the low pressure side, as a result of lower evaporation temperatures. As a consequence the chilled water temperature difference between the evaporator inlet and the outlet was increased, as shown in Figure A.1.

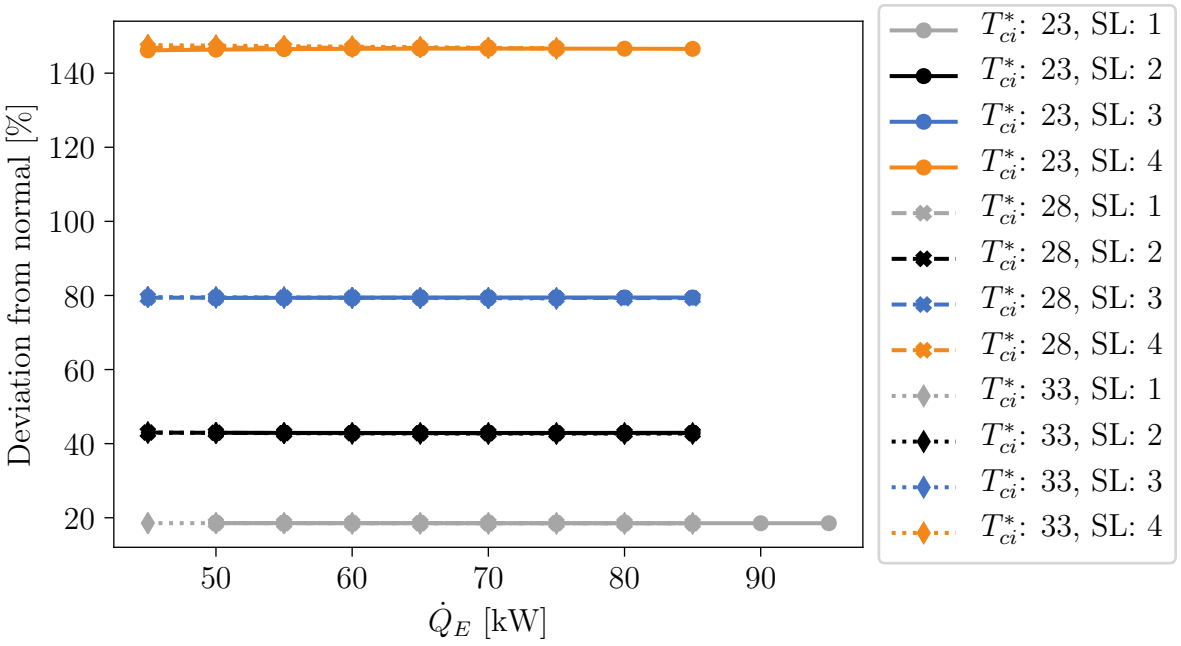

Figure A.1: Deviation of ΔT_E for RVE

A.4.2 Reduced Condenser Water-Flow Rate

The fault RVC was primarily characterised by a higher cooling water temperature difference between the condenser inlet and the outlet as well as a increased pressure within the system's high-pressure line. Yet, the condenser approach temperature was also significantly affected, as shown in Figure A.2.

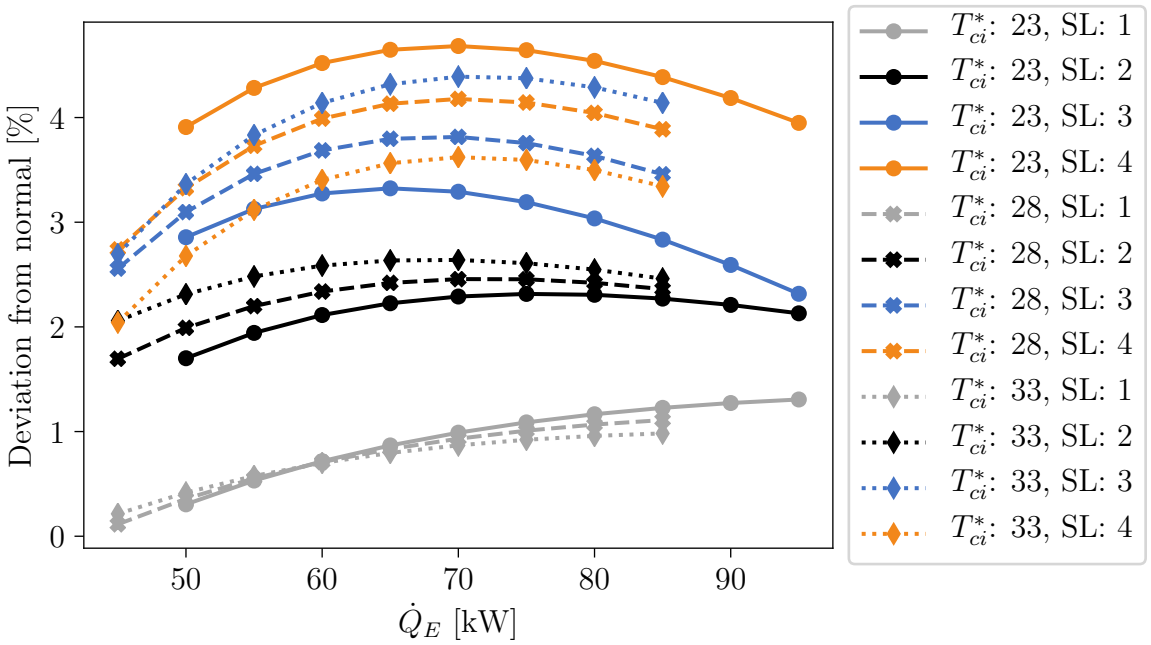

Figure A.2: Deviation of ΔT_{ca} for RVC

A.4.3 Non-Condensables

The most distinct fault pattern occurred for non-condensables gases in the refrigerant line (NC). In fact, the risen pressure within the high-pressure line indicated a significant change in the operation conditions, which was already clearly recognisable at low SL. Interestingly, the superheat discharge temperature decreased with increasing fault severity, as shown in Figure A.3. However, this statement is deceptive because this behaviour is caused by the oil temperature control loop (see Figure 4.1). Nevertheless, among other things, it is an example of the confirmation of Hypothesis H1.1. A further distinct

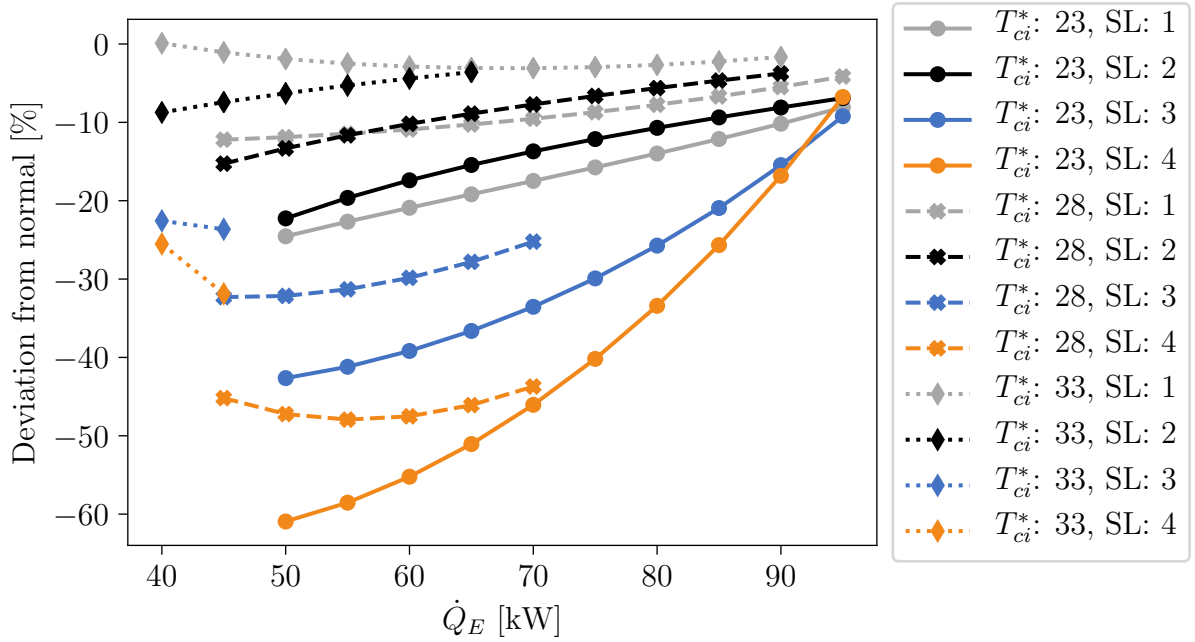

Figure A.3: Deviation of $\Delta T_{sh,dis}$ for NC

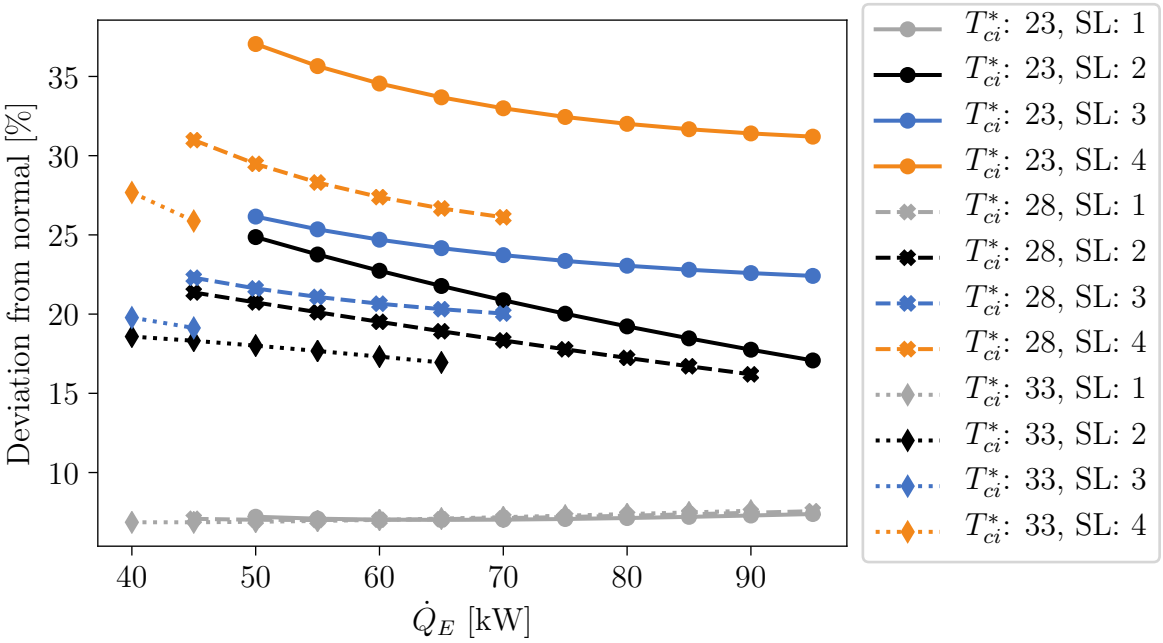

Figure A.4: Deviation of P_{comp} for NC

characteristic results from the increased pressure difference between the low-pressure and high-pressure sides of the refrigerant cycle. This increases the torque on the drive shaft of the screw compressor, which in turn increases the drive power, as shown in Figure A.4.

A.4.4 Refrigerant Leak

The last fault investigated throughout the data acquisition phase was RL. Justified by the design of the flooded evaporator, the data basis collected from these experiments is

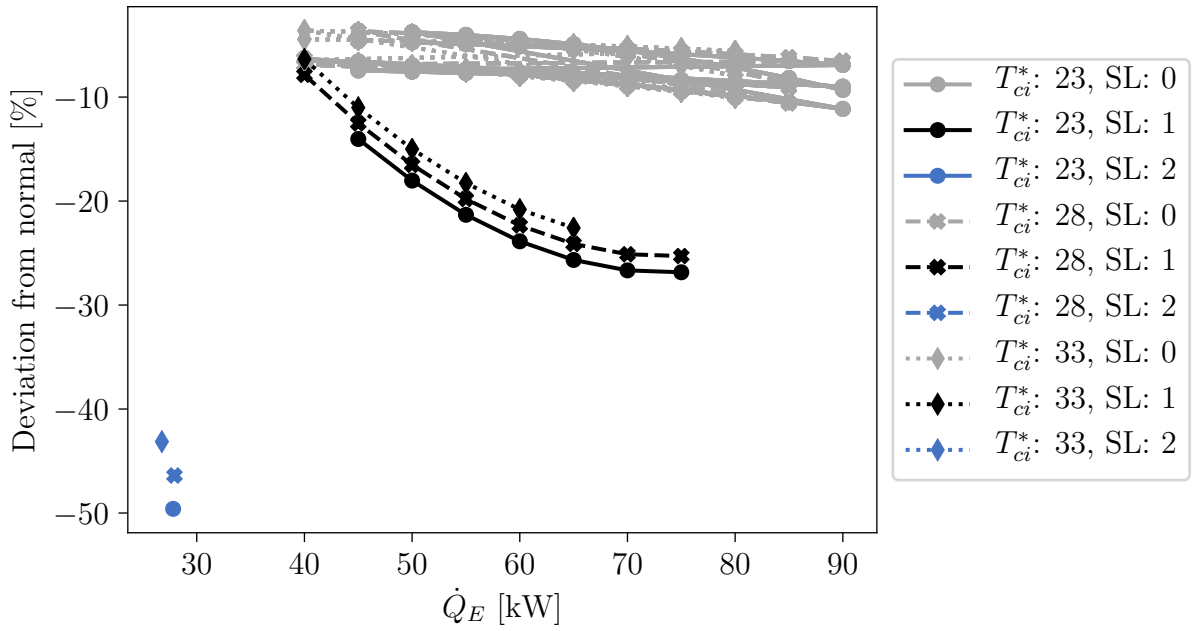

Figure A.5: Deviation of p_{re} for RL

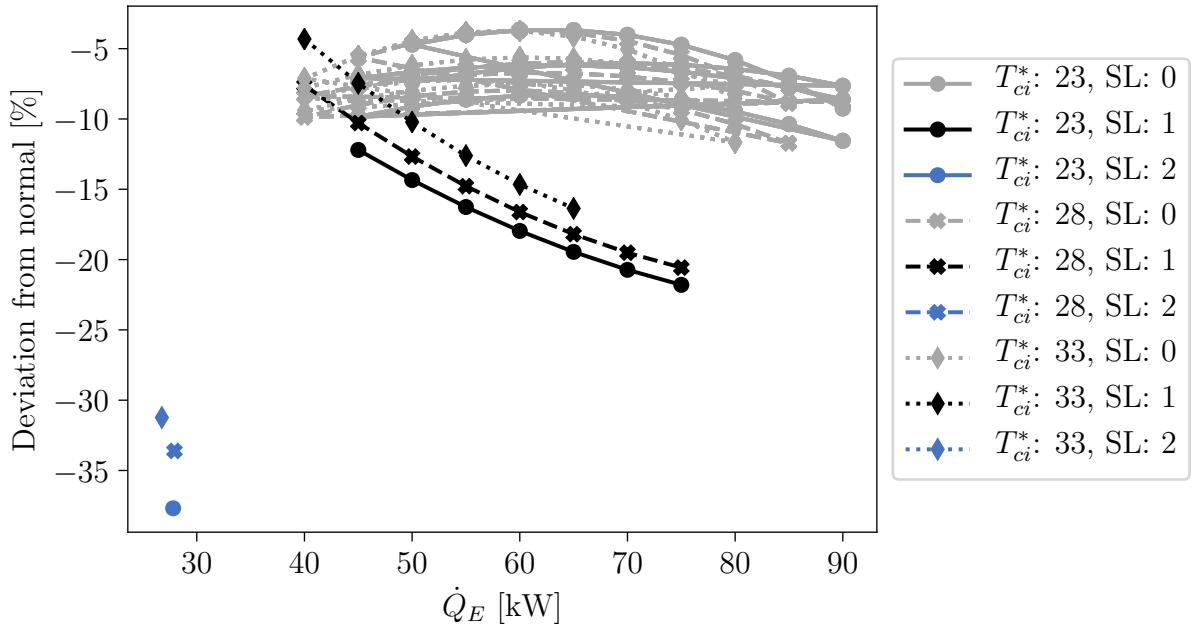

Figure A.6: Deviation of ε for RL

present in less detail than in the defects discussed previously. This matter was discussed in greater detail in Section 4.4.5. Yet, as shown in Figure A.5 using the example of the pressure in the evaporator, this fault also occurs with strongly pronounced fault patterns. As a consequence, the power consumption is increased and, thus, affects coefficient of performance, as shown in Figure A.6.

A.4.5 Average Deviation of Refrigerant Leak

As aforementioned, due to the chiller design used within the test rig, in particular the flooded evaporator design, the fault RL was more difficult to be examined. This is mainly due to the fact that fault characteristics are expected to be recognisable after the refrigerant filling quantity fell below a certain limit. As a result, the first test cases did not show well distinguishable fault patterns, which has been discussed in Section 4.4.5. For completeness, however, the results of all experiments related to this fault are listed in Table A.3.

Table A.3: Average deviations of all datasets associated with the investigation of refrigerant leak compared to the benchmark dataset.

Variable	RL0-1		RL0-2		RL0-3	
	δ_e	R^2	δ_e	R^2	δ_e	R^2
P_{comp}	7.37 %	0.996	8.35 %	0.996	10.52 %	0.995
p_{re}	-5.19 %	0.997	-7.95 %	0.996	-8.43 %	0.996
p_{rc}	1.44 %	0.999	1.08 %	0.999	1.26 %	0.999
$\Delta T_{sh,suc}$	35.64 %	0.879	46.19 %	0.815	57.08 %	0.737
$\Delta T_{sh,dis}$	15.60 %	0.881	16.24 %	0.911	17.28 %	0.886
ΔT_{ea}	73.49 %	0.984	119.58 %	0.943	126.44 %	0.978
ΔT_{ca}	3.16 %	0.998	1.54 %	0.997	2.07 %	0.997
ΔT_E	-0.07 %	0.999	-0.04 %	0.999	0.44 %	0.999
ΔT_C	9.25 %	0.995	8.28 %	0.995	9.35 %	0.995
ε	-6.62 %	0.963	-7.34 %	0.966	-8.70 %	0.972
T_{oil}	5.63 %	0.988	6.33 %	0.984	6.63 %	0.987
	RL0-4		RL1		RL2	
	δ_e	R^2	δ_e	R^2	δ_e	R^2
P_{comp}	6.80 %	0.996	19.08 %	0.988	49.82 %	0.990
p_{re}	-5.59 %	0.998	-19.28 %	0.965	-44.32 %	0.990
p_{rc}	0.99 %	0.998	0.79 %	0.998	0.80 %	0.999
$\Delta T_{sh,suc}$	41.45 %	0.911	238.48 %	0.905	468.97 %	0.910
$\Delta T_{sh,dis}$	19.41 %	0.963	21.67 %	0.930	78.52 %	0.934
ΔT_{ea}	79.04 %	0.992	323.52 %	0.954	1,422.07 %	0.997
ΔT_{ca}	1.11 %	0.998	1.73 %	0.994	0.69 %	0.930
ΔT_E	0.53 %	0.999	0.12 %	0.999	-0.38 %	0.999
ΔT_C	8.45 %	0.996	4.81 %	0.991	14.11 %	0.880
ε	-5.30 %	0.964	-14.70 %	0.969	-33.24 %	0.990
T_{oil}	5.82 %	0.989	6.44 %	0.993	12.95 %	0.998

A.5 Dataset Features

Table A.4 shows the features of the two domains selected for the dataset. As outlined previously, some features mentioned in Section 4.2.2 are discarded, which include the measured water flow rates as well as the ambient temperatures. Besides, the instantaneous motor current I_{comp} has not been particularly considered, as this information is already aggregated in the motor power consumption P_{comp}. Similarly, the compressor rotational speed n_{comp} obviously does not provide any fault indication, which has therefore not been considered. This also applies to the redundant sensor readings presented in Table 4.3.

Table A.4: Overview of the used and domain shared features.

Symbol	Designation
T_{ci}	Water temperature at condenser inlet
T_{co}	Water temperature at condenser outlet
T_{ei}	Water temperature at evaporator inlet
T_{eo}	Water temperature at evaporator outlet
T_{oil}	Oil feeding temperature
T_{suc}	Refrigerant suction temperature
T_{dis}	Refrigerant discharge temperature
p_{rc}	Refrigerant pressure at compressor outlet
p_{re}	Refrigerant pressure at compressor inlet
P_{comp}	Instantaneous motor power (compressor)
$\dot{Q}_C$	Condenser heat and heat flow
$\dot{Q}_E$	Evaporator heat and heat flow
T_{re}	Refrigerant evaporation temperature
T_{rc}	Refrigerant condensing temperature
ΔT_{ea}	Evaporator approach temperature
ΔT_{ca}	Condenser approach temperature
ΔT_E	Evaporator water-glycol temperature difference
ΔT_C	Condenser water-glycol temperature difference
$\Delta T_{sh,dis}$	Superheat temperature at compressor discharge
$\Delta T_{sh,suc}$	Superheat temperature at compressor suction line
ε	Coefficient of performance

A.6 Results from ROC Analysis

Following Section 5.3.1, the remaining results from ROC analysis are presented in hereinafter. Table A.5 shows the achieved AUC scores for the observed classifiers in dependence on the respective fault SL, the values of which are further graphically underlined through the ROC curves illustrated in the figures A.7 - A.9. The results also indicate the expected behaviour for faults occurring at higher SL, as has been discussed in Section 3.7.

Table A.5: The AUC values obtained from ROC analysis separately by severity and classifier.

Fault detection model	Fault	AUC Scores							
		Domain A				Domain B			
		SL=1	SL=2	SL=3	SL=4	SL=1	SL=2	SL=3	SL=4
BSVM	NC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	RVC	0.97	1.00	1.00	1.00	0.99	1.00	1.00	1.00
	RVE	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	RL	0.72	0.86	0.97	0.99	1.00	1.00	-	-
OC-SVM	NC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	RVC	0.98	1.00	1.00	1.00	0.95	1.00	1.00	1.00
	RVE	0.95	0.99	1.00	1.00	1.00	1.00	1.00	1.00
	RL	0.76	0.73	0.87	0.98	1.00	1.00	-	-
SPE	NC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	RVC	0.95	0.99	1.00	1.00	0.95	1.00	1.00	1.00
	RVE	0.92	0.99	1.00	1.00	1.00	1.00	1.00	1.00
	RL	0.80	0.85	0.90	0.99	1.00	1.00	-	-

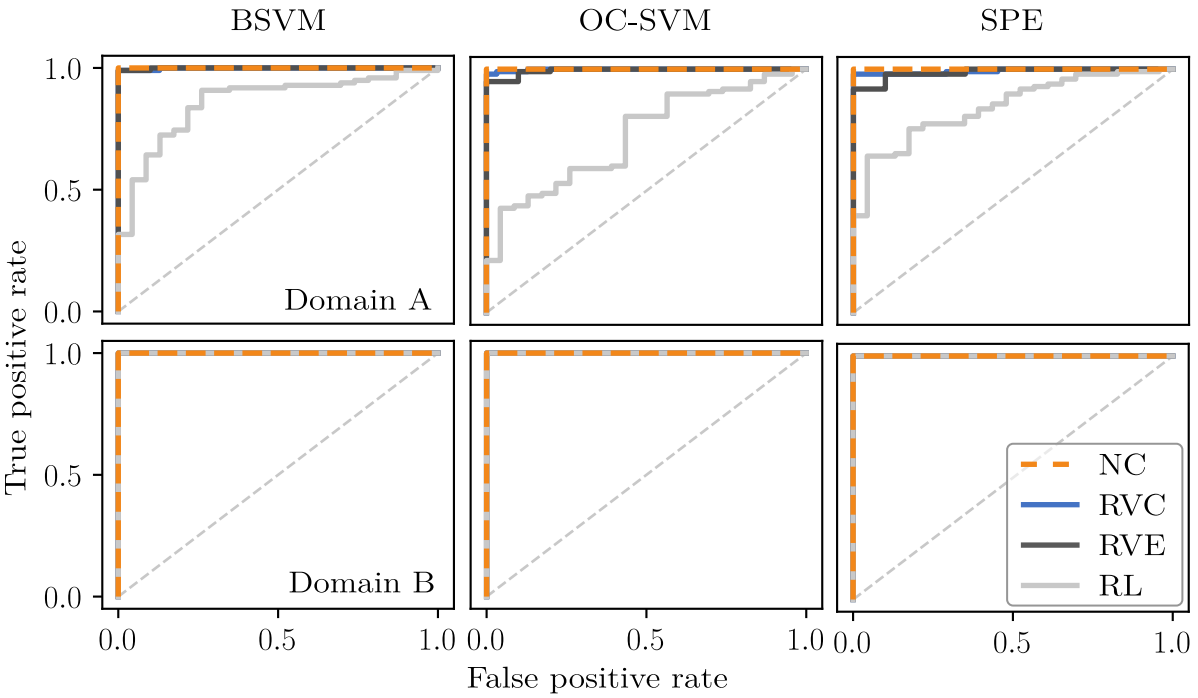

Figure A.7: ROC curves at SL=2

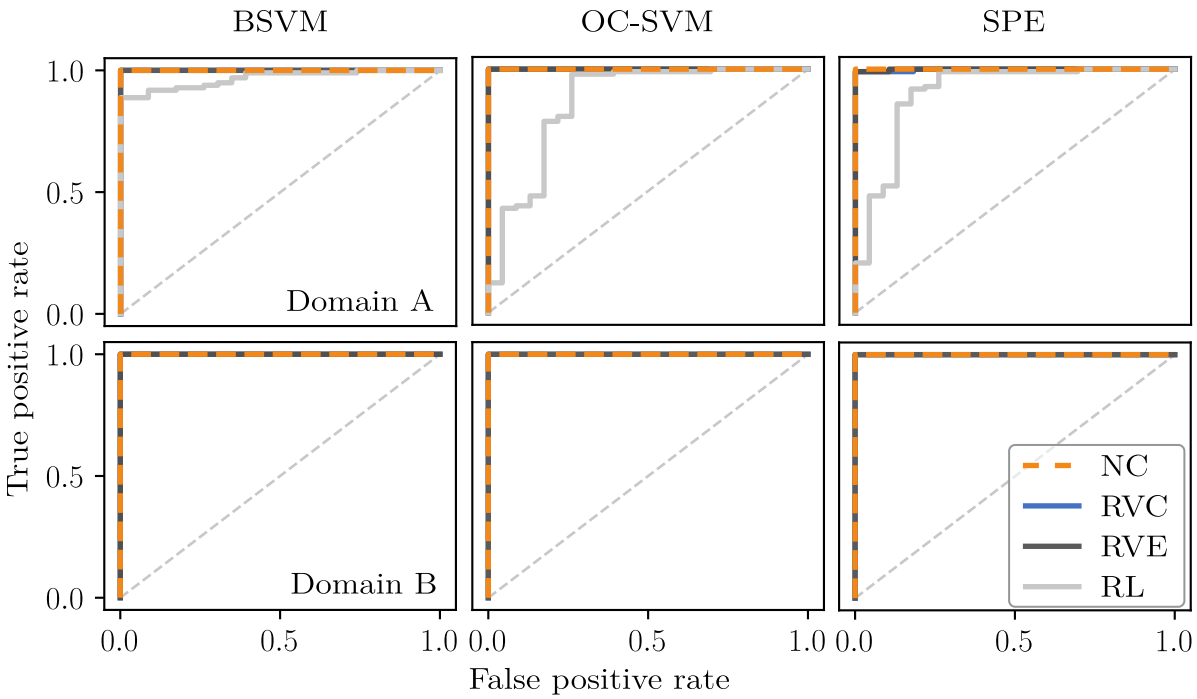

Figure A.8: ROC curves at SL=3

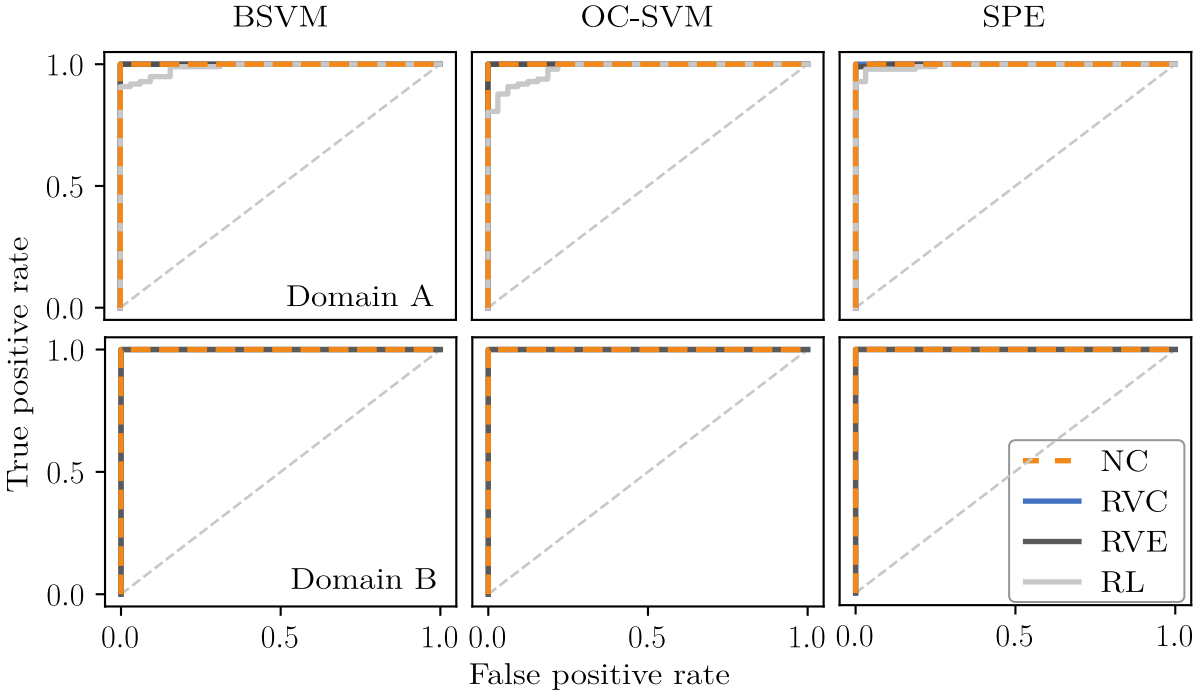

Figure A.9: ROC curves at SL=4

A.7 Feature Representation

This section provides an overview of the feature representation in the discussed domain adaptation problem. More specifically, it shows the aligned observations as a result of the models presented and discussed in Section 5.3.3. Thereby, LDA was applied to reduce the higher dimensional feature space to two dimensions for reasons of visibility. Table A.6 assigns the plots presented hereinafter to the respective model. The figures presented subsequently illustrate the representations of the two domains, whereas Figure A.10 shows the case Domain A → B and Figure A.11 Domain B → A. It is important to emphasise that the figures show the domain adaptation processes after reducing the process variability using PCA (at $k_{pc} = 7$), except for plot (a), which shows the original input spaces.

Table A.6: Scatter plots showing the LDA transformed feature representation.

Label	Model	Label	Model
a	No domain adaptation	e	PCA(X_t)
b	PCA(X_s), PCA(X_t)	f	TCA
c	PCA(X_s)	g	CORAL
d	PCA(X_s, X_t)	h	SA

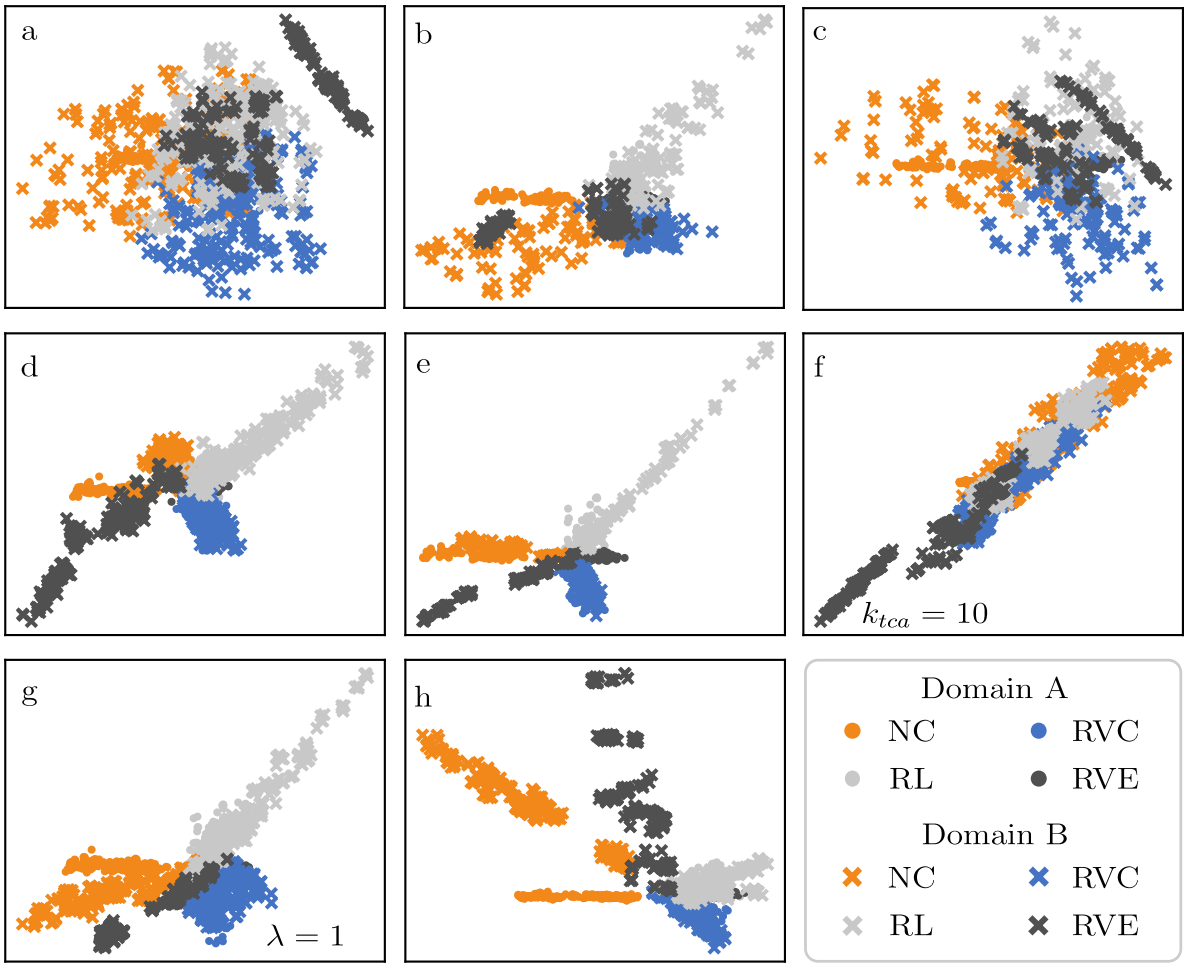

Figure A.10: Scatter plot for the task Domain A → B

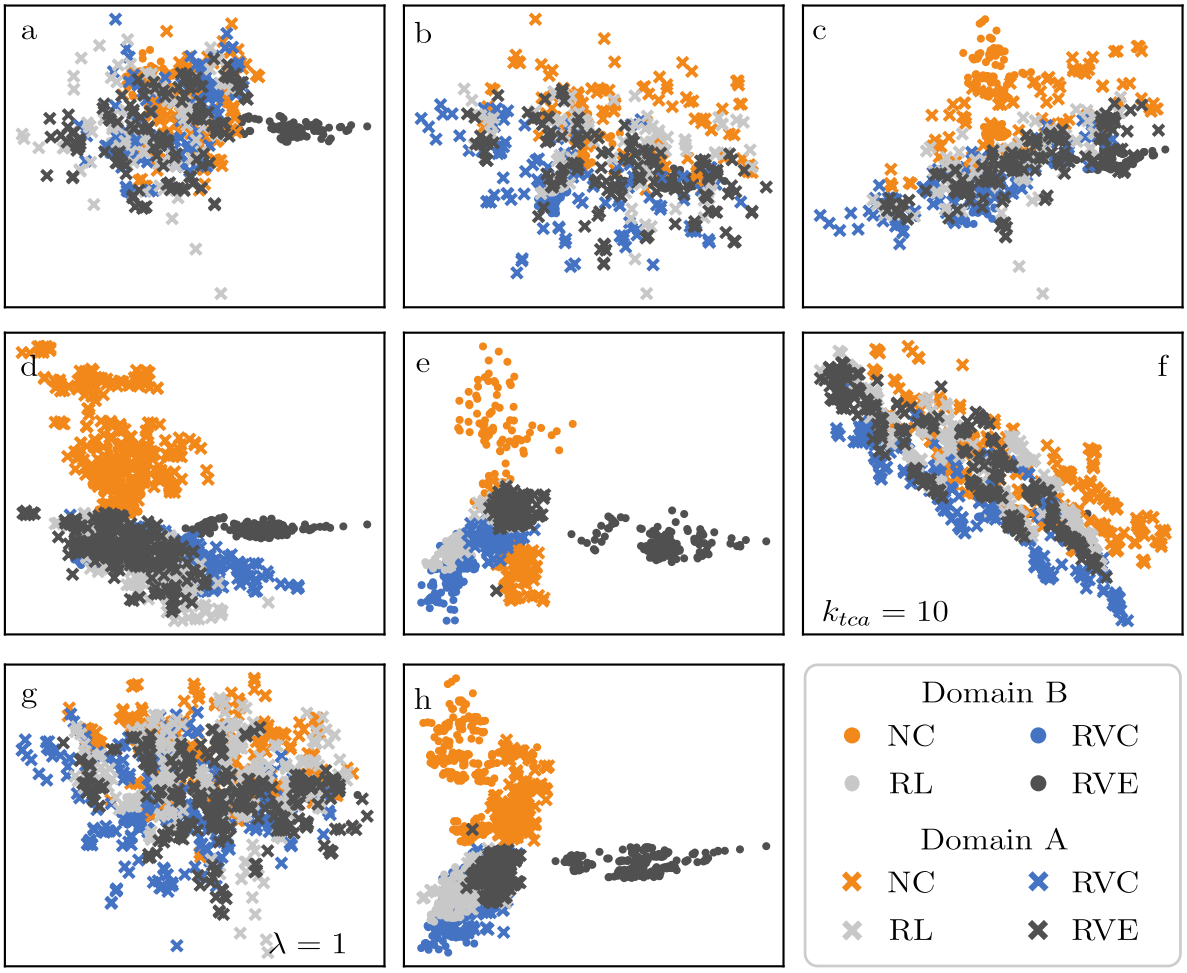

Figure A.11: Scatter plot for the task Domain B → A