Automatic fault detection and diagnosis in refrigeration systems, A data-driven approach

Soltani, Zahra

DOI (link to publication from Publisher):
10.54337/aau535542944

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

AUTOMATIC FAULT DETECTION AND DIAGNOSIS IN REFRIGERATION SYSTEMS, A DATA-DRIVEN APPROACH

BY
ZAHRA SADAT NOURBAKHSH SOLTANI

DISSERTATION SUBMITTED 2023

AALBORG UNIVERSITY
DENMARK

Automatic fault detection and diagnosis in refrigeration systems, A data-driven approach

Ph.D. Dissertation

Zahra Sadat Nourbakhsh Soltani

Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7C
DK-9220 Aalborg

Dissertation submitted: February 2023

PhD supervisor: Associate Professor John-Josef Leth
Aalborg University

PhD committee: Professor Roozbeh Izadi-Zamanabadi (chair)
Aalborg University, Denmark

Automation Engineer Christoph Josef Backi
BASF SE, Germany

Associate Professor Hamid Reza Shaker
University of Southern Denmark, SDU, Denmark

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Electronic Systems

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-742-0

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Stibo Complete, 2023

In this work, data is acquired from both a real refrigeration system installed in Bitzer electronics and a high-fidelity model in *Matlab/Simscape* which has been used in the company for multiple purposes such as controller development. The real data is acquired from an embedded system called *LMC* which is a controller produced by Bitzer electronics.
Data pre-processing are done in *Python* using *pandas* library, and Machine Learning models are made in *Python* using *sklearn* library which provides several machine learning toolboxes for the users. This document is written in Latex environment. The plots are printed in Python and self-made figures are sketched in Microsoft *Visio*.

Abstract

This study is concerned with the capability of machine learning models in fault detection and diagnosis (FDD) in refrigeration systems such as supermarket refrigeration systems. The functionality of refrigeration systems has huge impacts on energy consumption, preserving goods in cold storage, and decreasing air pollution. Thus, the development of FDD tools in this industry is desired by manufacturers, customers, and environmental protection agencies. The goal is to investigate FDD algorithms that perform satisfactorily and are robust regarding the accuracy, false positive rate, and applicability.

This study compared a deep learning algorithm called Convolutional Neural Network (CNN) and multiple shallow learning algorithms such as Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA). In addition, Principal Component Analysis (PCA) and LDA are implemented for preprocessing of the data. In this regard, data from a refrigeration system that is set up in Bitzer Electronics is acquired, and further, simulation data is collected from a high-fidelity Simscape model which has been used in the company.

The results illustrate that the CNN model is highly robust in binary classification tasks when unseen data is noisy or perturbed, and the number of data samples affects the model's accuracy. PCA-SVM performs better than SVM regarding classification accuracy and computation time. PCA reduces input dimensions to only two and helps the SVM classifier deal with noise, perturbation, and running the system in on/off mode. Data with different sample rates in the range [1,0.01] Hz does not affect the mentioned models' accuracy in the classification task. For multi-class classification purposes, SVM and LDA-SVM perform better than CNN, LDA, and PCA-SVM. SVM and LDA-SVM can identify 18 classes out of 21 classes with 100% accuracy, while the LDA-SVM stands alone for detecting faulty systems from non-faulty systems with a 0% false positive rate. This means that LDA-SVM is a reliable model for fault detection purposes, and SVM performs better than the other models for fault localization.

Resumé

Denne undersøgelse beskæftiger sig med nøjagtigheden af maskinlæringsmodeller inden for fejldetektion og -diagnose (FDD) i kølesystemer såsom supermarkedskølesystemer. Funktionaliteten af et køleanlæg har stor indflydelse på energiforbruget, konservering af varer i kølerum og mindskelse af luftforurening. Udviklingen af FDD-værktøjer i denne industri er således efterspurgt af producenter, kunder og miljøbeskyttelsesagenturer. Målet er at undersøge FDD-algoritmer, der fungerer tilfredsstillende og er robuste med hensyn til nøjagtighed, falsk positiv rate og anvendelighed.

Denne undersøgelse sammenlignede en dyb læringsalgoritme kaldet Convolutional Neural Network (CNN) og flere overfladiske læringsalgoritmer såsom Support Vector Machines (SVM) og Linear Discriminant Analysis (LDA). Derudover er Principal Component Analysis (PCA) og LDA implementeret til forbehandling af dataene. I den forbindelse indhentes data fra et køleanlæg, som er opsat hos Bitzer Electronics, og yderligere indsamles simuleringsdata fra en high-fidelity Simscape-model, som er blevet brugt i virksomheden.

Resultaterne illustrerer, at CNN-modellen er meget robust i binære klassifikationsopgaver, når usete data indeholder støj eller forstyrrelser, og antallet af samples påvirker modellens nøjagtighed. PCA-SVM yder bedre end SVM med hensyn til klassificeringsnøjagtighed og beregningstid. PCA reducerer inputdimensioner til kun 2 og hjælper SVM-klassifikatoren med at håndtere støj, forstyrrelser og at systemet kører i on/off-tilstand. Data med forskellige sample rates i området [1,0.01] Hz påvirker ikke de nævnte modellers klassificeringsnøjagtighed. Til flerklasseklassifikationsformål klarer SVM og LDA-SVM sig bedre end CNN, LDA, og PCA-SVM. SVM og LDA-SVM kan identificere 18 klasser ud af 21 klasser med 100% nøjagtighed. LDA-SVM er bedst til at detektere fejlbehæftede systemer fra ikke-defekte systemer med en 0% falsk positiv rate. Det betyder, at LDA-SVM er en pålidelig model til fejldetektionsformål, og SVM yder bedre end de andre modeller til fejllokalisering.

Contents

Thesis Details

Thesis Title:	Automatic fault detection and diagnosis in refrigeration systems, A data driven approach
PhD. Student:	Zahra Sadat Nourbakhsh Soltani
Supervisors:	Assoc. Prof. John Leth, Aalborg University Assoc. Prof. Jan Dimon Bendtsen, Aalborg University PhD Kresten Kjær Sørensen, Bitzer electronics

The main body of this thesis consist of the following papers.

[A] Zahra Soltani, Kresten Kjær Sørensen, John Leth, Jan Dimon Bendtsen, "Fault Detection of Supermarket Refrigeration Systems Using Convolutional Neural Network," *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 231-238, 2020.

[B] Zahra Soltani, Kresten Kjær Sørensen, John Leth, Jan Dimon Bendtsen, "Robustness analysis of PCA-SVM model used for fault detection in supermarket refrigeration systems," *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1-6, 2021.

[C] Zahra Soltani, Kresten Kjær Sørensen, John Leth, Jan Dimon Bendtsen, "Fault detection and diagnosis in refrigeration systems using machine learning algorithms" *International Journal of Refrigeration*,
doi: https://doi.org/10.1016/j.ijrefrig.2022.08.008, vol. 144, pp. 34-45, 2022. issn: 0140-7007.

This dissertation has been submitted for assessment of the PhD degree. This PhD study was an industrial PhD, and it looked into the problem with a commercial focus. The thesis is based on the collection of published scientific papers which are listed above. The contents of the papers are used directly or indirectly in the extended summary of this thesis. Co-author statements are available at the doctoral school as well as to the assessment committee as a requirement for the assessment.

Preface and acknowledgements

This dissertation is to fulfil the requirements for my doctoral research degree. The thesis is based on the collection of the research conducted during the PhD program at the Section of Automation and Control, Department of Electronic Systems at Aalborg University, Denmark. Here, I have an excellent opportunity to appreciate my supervisors, Assoc. Prof John-Josef Leth and Assoc. Prof Jan Dimon Bendtsen, who gave me the freedom to determine the research direction and stayed with me in the whole PhD course, advised me and helped me with all academic aspects and personal challenges that a PhD student may concern about.

The research was offered in April 2019 by Bitzer Electronics Denmark and funded by Innovation Fund Denmark. The course of the PhD was challenging with all ups and downs, combined with the difficulties of the Corona pandemic. Fortunately, I was not alone on this road. I express my deepest gratitude to my co-supervisor, PhD. Kresten Kjær Sørensen. He was full of knowledge and ideas, keen on sharing them, and always available, despite his busy schedule. Special thanks to Mads Philipsen, R&D Director at Bitzer Electronics for giving me the opportunity to put this study into practice.

I am grateful for doing this research among many expert colleagues in Bitzer Electronics and Aalborg University who are involved in this research with their ideas and comments.

Many thanks to my husband Faramarz, and our children, Ava and Sina, for their love, patience, and encouragement. I would have never reached this dissertation's end without their support. I would also like to thank my brother Mohsen who inspired and motivated me at the beginning of the PhD program.

Last but not least, I send a thank with all my love to my parents from far away for their patience and support. They may know that their big smiles for any of my tiny achievement have always been the big motivation to take another challenging step.

Zahra Soltani
Aalborg University, Denmark February 10, 2023

Nomenclature

Latin symbols

A	cross sectional area of the compressor inlet (m^2
b	bias
C	regularization parameter for ζ
c	number of feature vectors
$Ctrl$	Controller
d	order of polynomial function
f	activation function
H	hyper plane
I	current (A)
L	loss function
l	leyer
M	molar
MP	size of maxpooling
n	number of samples
NF	number of filters
P	pressure (Pa)
p	epoch
p	power (W)

R	ideal constant of gas refrigerant
r	transformed sample
S	number of the neurons in the last layer
S	variance
SF	size of filter
T	temperature (°C)
V	matrix of eigenvectors
V	refrigerant specific volume ($\frac{m^3}{Kg}$)
V	speed (Hz)
v	eigenvectors
$vexp$	expansion valve opening degree (%)
W	weight matrix in CNN
W	work (J)
w	weight vector
X	input
x	input sub-matrix in CNN
x	input vector in SVM and LDA
Y	output of each hidden layer
y	output of the neuron

Greek symbols

α'	number of selected eigenvectors
α	learning rate
χ	feature vector
γ	GRBF multiplier
λ	eigenvalue

μ	mean
∇	gradient
ν	number of classes
Ω	transformation matrix
Φ	GRBF function
ρ	density in suction line ($\frac{kg}{m^3}$)
$\underline{\Delta}$	Disturbance
ζ	slack variable

Subscripts and superscripts

0	evaporation
$\dot{m}$	mass flow rate ($\frac{Kg}{s}$)
$\hat{}$	predicted
κ	κ_{th} class
ϕ	ϕ_{th} vector
θ	influenced by disturbance
amb	ambient
C	condensing
$cond$	condenser fan
cpr	compressor
dis	discharge
$evap$	evaporator fan
$fault$	value after applying the fault
G	between-class
hx	heat exchenge
i	compressor inlet

j	j_{th} sample
o	compressor outlet
$offset$	value for fault emulation
$real$	value before applying the fault
ret	returned air
$room$	cooling room
S	number of weight matrices in CNN
s	within-Glass
sh	superheat (°K)
suc	suction gas
sup	supplied air

Abbreviations

ANN	Artificial Neural Network
AUC	Area Under the Curve
BP	Back Propagation
CNN	Convolutional Neural Network
DC	Direct current
$DRNN$	Deep Recurrent Neural Network
DTI	Deep Transient Input
DTO	Deep Transient Output
ELU	Exponential Linear Unit
FC	frequency converter
FDD	Fault Detection and Diagnosis
FF	Feed Forward
GA	Genetic Algorithm

GB Gradient Boosting

$GRBF$ Gaussian Radial Basis Function

$HVAC\&R$ Heating, Ventilation, Air conditioning & Refrigeration

$HVAC$ Heating, Ventilation, Air conditioning

IoT Internet of Things

KDA Kernel Discriminant Analysis

KDE Kernel Density Estimation

LDA Linear Discriminant Analysis

$LSTM$ Long-Short Term Memory

ML Machine Learning

MLP Multi Layer Perceptron

OVO One Versus One

OVR One Versus Rest

PCA Principal Component Analysis

$ReLU$ Rectified Linear Unit

RF Random forest

ROC Receiver Operating Characteristic

RS Refrigeration Systems

$S, DTO - DRNN$ Stacked and DTO-DRNN

$S - DRNN$ Stacked-DRNN

SRS Supermarket Refrigeration Systems

$SVDD$ Support Vector Data Description

SVM Support Vector Machine

$Tanh$ Tangent hyperbolic

Part I

Introduction

Extended summary

1 Motivation

A cold chain is a low-temperature controlled chain from the point of production or manufacturing to the end consumers, shown in Fig. 1. Perishable food, beverage, pharmaceuticals, and temperature-sensitive chemicals are preserved in the cold chain. refrigeration systems play an important role in the cold chain as they control the temperature of the goods' environment. Refrigeration systems are required in manufacturing and transporting products, warehouses, supermarkets, pharmacies, and households [1]. Even a small unexpected change in the cold storage temperature may have a huge impact in different areas as follow:

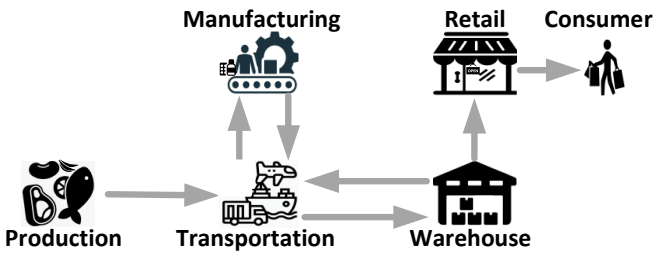

Fig. 1: An overview of a cold chain from production and manufacturing to the consumers where in each section, the functionality of refrigeration systems is essential to ensure the goods are delivered safely to the end user.

- Human health
  A cold chain can affect food nutrition, and improper cooling processes in the cold chain may lead to some irreversible consequences in societies. For instance, it can potentially affect a society's population's height in the long term [2], foodborne

diseases, and so on. Therefore, minimising the risk of such consequences is demanded by standardising the cold chain and monitoring the product's quality [3].

- Medicine
  For many decades, sensitivity and accuracy of storage temperature in refrigeration systems have been considered for protecting medical products and developments [4]. During the corona pandemic, refrigeration systems were essential for vaccine storage and distribution to rescue the world's population. [5]. Therefore, temperature control and monitoring are vital to the security and development of medical solutions.

- Global warming
  Refrigeration systems are one of the products whose functionality is important for environmental agencies. Additional energy consumption due to a refrigeration systems failure, air pollution/earth pollution due to the waste of products, and refrigerant leakage or other refrigeration systems failures impact the environmental issues [3].

- Economy
  About 30% of all produced food is annually wasted due to inappropriate food preservation. For example, in the USA, $218 billion is spent yearly on the cold chain for food that is never consumed. In the EU, around 10% of the imported food is wasted, which is roughly €6.63 billion per year [1]. Thus, the preservation of food appropriately saves money since most of the produced food is consumed, and finally, demands on the amount of food production are reduced.

Today, the responsible authorities for each item mentioned above are aware of the importance of cold chain performance. Therefore, they request manufacturers pay more attention to the accurate and robust storage temperature and continuous monitoring to detect any failure at an early stage. From the users' point of view, early fault detection in refrigeration systems is cost-saving considering decreasing power consumption, maintenance cost, fatigue faults' cost, and spoilage of goods.

Bitzer Electronics produces HVAC system controllers and supplies cloud-connected electronic protection modules called IQ modules. IQ modules are integrated into the compressors produced by Bitzer in Germany. IQ modules have basic fault detection capabilities to identify some severe faults that destroy the systems. In addition, technicians handle the refrigeration system monitoring at a fixed interval or upon request. In many refrigeration systems, faults happen gradually, and performance is affected over time. This means that a failing system often runs for an extended period at low efficiency before the error is discovered. Bitzer Electronics is interested in a solution to increase its products' uptime, efficiency, and reliability. Early fault detection and diagnosis in refrigeration systems are proposed to improve the competitiveness of the products. Due

to system configurations and component variations, data-driven approaches are investigated in this work. This study takes advantage of looking at the problem from an industrial perspective. The investigations during this study can improve other business areas at Bitzer Electronics, such as ventilation systems and heat pumps and turn into a core technology for the company, which in time may help improve most of the products at Bitzer Electronics.

2 Research Question

This chapter addresses the problem of fault detection and diagnosis that exists in the refrigeration industry and the challenges this research covers.

The problem of automatic and data-driven FDD applications in refrigeration systems appeared some decades ago when data availability increased. Running a malfunctioning refrigeration system may lead to various components damages in the short or long term, low efficiency, or high power consumption.

Bitzer Electronics A/S produces controllers which are widely used in HVAC&R applications. In addition, the parent company Bitzer provides so-called "condensing units", in which components for compression and condensation in the refrigeration cycle are mounted. Therefore, in many applications, for instance, supermarket refrigeration systems, Bitzer Electronics has access to data from a part of the refrigeration systems. However, since Bitzer provides condensing units, there is usually no information about evaporation and expansion phases; the numbers and sizes of evaporators, evaporator fans, cold rooms, expansion valves, and the design of evaporation-side controllers differ from application to application.

In other applications, on the other hand, the controller produced by Bitzer Electronics controls the whole refrigeration systems, such as heat pumps, reefer containers, and reefer trucks.

It is thus only natural that Bitzer Electronics is interested in robust solutions that are able to work effectively in vastly different operating conditions and configurations.

The research undertaken in this PhD study addresses the questions below:

- Is it possible to find faults in the evaporation side of the refrigeration system using data available to Bitzer, such as data from the inlet of the compressors?
- Is it possible to use machine learning methodologies to simultaneously identify several components and sensor faults in refrigeration systems?
- Do deep learning classifiers perform better than shallow learning classifiers?
- How many data samples are required for fault detection and diagnosis in refrigeration systems using selected classifiers?
- Comparison of the classifiers regarding computation time, accuracy and false positive rate.
- Does sample rate affect the classification accuracy?
- Are the selected classifiers robust in different operational conditions?
- Are the selected classifiers robust against noise?

- How do the selected classifiers perform when the refrigeration system alternate between on and off mode?
- What is the effect of training data excitation or variation on training the models?
- Does any data feature affect the classification results inversely?

Finding answers to the questions above is valuable to Bitzer Electronics, as it would allow the company to introduce more efficient, reliable and intelligent products and increase components' uptime due to early fault detection across entire classes of refrigeration systems. The challenges above are discussed in Sections 7 and 8, and the answers can be found in Section 9.

According to the considerations above, different data collection scenarios are considered in the research reported in this thesis. Papers A and B consider data from a condensing unit without information about the evaporation and expansion phases, whereas the data acquired in Paper C contains information from a complete refrigeration system. Paper C studies some classifiers that identify several possible faults in refrigeration systems, including sensors and components faults.

3 Content outline

This report consists of two parts. Part I presents the extended summary of the published papers which are done during the PhD study. Chapter 1 explains the importance of the research from different aspects, such as human health, global warming and the economy. The problem of fault detection and diagnosis in refrigeration systems and the reasons for carrying this research out are argued in Chapter 2. In Chapter 4, the refrigeration systems background, description of potential faults in refrigeration systems, the test setups configuration are presented. Recent fault detection and diagnosis studies are introduced in Chapter 5. In this chapter, state-of-the-art is investigated in Heating, Ventilation, Air conditioning, and refrigeration systems. Afterwards, the methods used in Papers A, B, and C are introduced in Chapter 6. In addition, several experiments are introduced in Chapter 7, which are conducted during the study and are not mentioned in the papers. In Chapter 7, description of the faults emulated or simulated in this study, data acquisition specification, and data visualisation are also presented. Then, the results achieved in the papers are summarized in Chapter 8. The summary of the main contributions of Papers A, B, and C are discussed in Chapter 9. Finally, practical remarks of the papers are concluded in Chapter 10.

Part II consists of three scientific articles which are conducted during this research. Papers A and B are studied based on the data from a real refrigeration system which is set up in the Bitzer electronics laboratory. These two research investigate different classifiers for binary classification between functional and faulty systems. Paper C compares and evaluates several fault detection and diagnosis models which can classify different types of faults in refrigeration systems.

Appendix 1 describes the important corrections in Paper C, which are investigated after publication of the paper. Therefore, The corrected sections of Paper C are applied to this report as an appendix.

4 Refrigeration systems

This chapter consists of the preliminaries in refrigeration systems, the problem of fault detection, and the refrigeration systems' models which are used in this study for data acquisition.

4.1 Refrigeration systems background

The purpose of using refrigeration systems is to transfer unwanted heat from a cold room to the ambient environment. Heat is defined as a transfer of energy between two regions which are not in the same thermal equilibrium [6]. Refrigeration systems have four primary components that fulfil thermodynamic behaviour during heat exchange: compressor, condenser, expansion valve, and evaporator. These main components are responsible for changing the refrigerant pressure, temperature and phase in a cycle shown in Fig. 2. Thermodynamic behaviour in vapour-compression refrigeration systems is described below:

- Compression
  In the compression process, the low-pressure, low-temperature gas refrigerant enters the compressor where the refrigerant is compressed to a high-pressure, high-temperature gas and delivered to the condensation phase, see Fig. 2. Note that if the ambient/coolant temperature and the refrigerant temperature are in equilibrium, heat transfer is not possible in the condensing phase. Therefore, the compressor is essential before condensation to increase the refrigerant temperature and pressure and ensure heat transferability to the ambient environment. Some different compressors are hermetic, reciprocating, open drive, semi-hermetic reciprocating, and scroll [7].

- Condensation/Heat dissipation
  During condensation, heat dissipation leads to a change of refrigerant phase from liquid to gas. A condenser releases heat to the ambient environment, such as air or water. Three processes happen during condensation while refrigerant pressure (P) is constant. First, the hot gas refrigerant at the inlet of the condenser is de-superheated. It means that the refrigerant temperature drops while the pressure is constant. In this part, 10-15% of total de-superheating is done. Second, heat dissipation leads to refrigerant condensation. Therefore, the refrigerant phase changes from gas to liquid. The last part of the condensation process is called subcooling, in which liquid refrigerant temperature decreases. Different condensers are air-cooled, evaporative water-cooled, shell and tube condensers, and plate heat exchangers [7]. Selection of the condenser in the refrigeration system depends on several factors such as weather conditions, availability of electricity, availability

of water, refrigeration system specification, heat capacity, refrigerant type, and operation period [6].

- Expansion
  The expansion valve is mounted just before the evaporator to reduce refrigerant pressure and temperature and make the refrigerant ready to vaporize in the evaporator pipeline. Therefore, 20-40% of the refrigerant evaporates during the expansion process to the pressure in the evaporator. The amount of vaporization depends on the operating condition of the refrigeration system and the requirements. In this process, changes in temperature and pressure happen due to the volume expansion of the refrigerant. There are different expansion valves, such as thermostatic, electronic, capillary tubes, and float valves. Thermostatic expansion valves work based on the thermodynamic behaviour of the refrigerant. In electronic expansion valves, the opening degree is controlled by the controller. Capillary tubes are usually used in small-size refrigeration systems, and the float valves are controlled based on the pressure at the high-pressure side [6].

- Evaporation/Heat absorption
  An evaporator is responsible for transferring heat from a cold room environment to the refrigerant. The refrigerant temperature at the inlet of the evaporator should be less than the cold room environment so that the refrigerant can absorb the heat. Heat absorption is an isobaric process in which a liquid phase changes to a gas. After the phase change, the gas refrigerant is superheated until it leaves the evaporator outlet while the pressure is constant. Evaporators can be found in two types, air or liquid-cooled [7].

Beside mentioned components, there are normally additional components which help a complete refrigeration cycle, such as condenser fan, evaporator fan, receiver, sensors, controller, et Cetra. shown in Fig. 3. These components are introduced below:

- Refrigerant
  Refrigerant is a fluid that circulates through the pipes and other components to absorb heat from an area with unwanted heat and release it to the outside. The type of refrigerant is selected based on the required cooling capacity, ambient temperature and refrigeration system design. The most common refrigerants that are currently in use are divided into three categories: *Halogenated hydrocarbon compounds* such as R134a and R32, *Organic hydrocarbon compounds* such as propane and isobutane, and *Inorganic compounds* such as CO_2 and Ammonia [7]. Usually, refrigerants have low boiling points and their volume increase by increasing the temperature.

- Frequency converters
  Compressors utilize mechanical energy in order to compress the gas flow. This

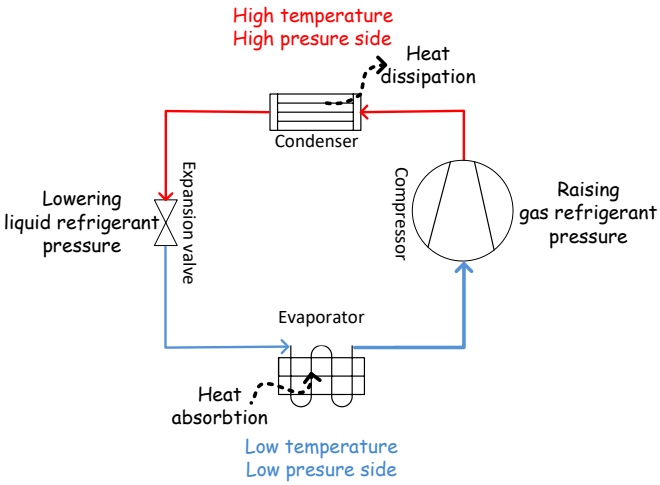

Fig. 2: Primary setup for a vapour-compression refrigeration system. The blue line indicates the low-pressure side and the red line indicates the high-pressure side of the refrigeration cycle.

energy is supplied by electricity. The compressors' motors change electrical energy to mechanical. The motors need specific voltage and frequency based on the compressors' capacity, size, and type of refrigerants. Frequency Converters (FC) are used to modulate the rated frequency to an adjustable frequency.

- Receiver
  Receivers are the containers located right after condensers. Receivers are used to ensure the required amount of refrigerant in each operating point. In other words, if the amount of refrigerant is more than required in an operating point, the condenser would not have sufficient space to store the refrigerant. Therefore, a receiver is needed to store condensed liquid refrigerant. In practice, a receiver is used if an operating point uses more than 3.6 Kg refrigerant [6].

- Evaporator fan
  Evaporators are generally located in the cold room. In the absence of a fan, cold room temperature can vary in different spots. Thus, an evaporator fan is required to make a consistent air temperature all over the cold room. In addition, the evaporator fan helps heat transfer from the air inside the cold room to the refrigerant. Evaporator fan speed can be determined by the controller such that the required heat transfer is done before the refrigerant leaves the evaporator.

- Condenser fan

The amount of heat accumulated in the refrigerant during evaporation and compression needs to be released to another environment. The refrigerant is condensed at a temperature above the ambient temperature around the condenser. A condenser fan is required to ensure that heat is released before the refrigerant leaves the condenser. The controller controls the fan speed.

- Controller
  Controllers control the components of refrigeration systems to ensure sufficient heat exchange during each operating point. Controllers are generally designed to control cold room temperature or suction pressure.

- Sensors
  Sensors are mounted in different positions in refrigeration systems. Sensors give the controller information about the refrigeration cycle or components' behaviour. Different sensors used in refrigeration systems can be temperature, pressure, vibration sensors, et Cetra.

- Liquid line valve
  Liquid line valves control the flow passing from condensing unit to the expansion valve. Liquid line Solenoid valves are used in refrigeration systems where a thermostatic expansion valve is used, or there is no information about the expansion valve and the evaporation side.

- Oil separator
  Compressor oil is used for the lubrication of pistons. It is usually mixed with refrigerant and passes through the discharge line. Therefore, an oil separator is required to separate and return the oil to the compressors. Oil separators are located in the discharge line (between the compressor and condenser). Note that between 0.0003% to 0.001% of the total amount of oil can not be separated and passes the refrigeration cycle along with the refrigerant.

- Filter dryer
  This filter is located after the receiver to deal with a tiny amount of existing moisture after installation of the system. The type and amount of refrigerant, line size, and allowable pressure drop affect the dryer selection.

- Sight glass
  To visualize that only liquid gas is delivered to the expansion, sight glass is utilized. However, sight glasses may show air bubbles due to several reasons. For example, there may be an insufficient refrigerant charge, pressure drop in the pipes, shut-off filters or devices, foreign gas like air or nitrogen in the system, or extreme heat transfer into the liquid line from surroundings [7].

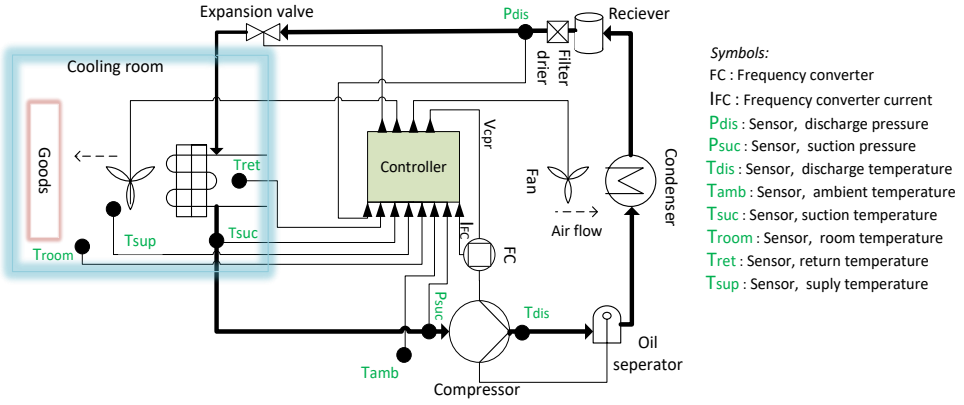

Fig. 3: Schematic of a refrigeration system including common components that help a complete and more efficient refrigeration cycle.

- High-pressure and low-pressure switches
  High-pressure and low-pressure switches are located before the compressor's inlet and after the compressor's outlet. These two switches are closed in any regular operation. A high-pressure switch works as an emergency switch when it is necessary to shut down the FC and stop the compressor. A Low-pressure switch is open when the pressure at the compressor's inlet is insufficient.

4.2 Faults in refrigeration systems

Fault, in general, is a term for malfunctioning a system or a component. Each system or component has its operation specification. An operation out of the specification limits is considered a faulty operation. Operating a faulty system may lead to stopping operation, damaging other components, or fatigue faults in the system. Thus, detecting, identifying, and fixing the fault has always been preferred to be done on time. Fault detection and diagnosis procedure can be done by technicians, technical tools, or automatically.

According to [8], failures in refrigeration systems are divided into four categories as follows:

- Component faults
  Malfunctioning refrigeration system components and sensors lead to different symptoms in the system. Some of the faults may also have the same symptoms, which

makes it more difficult to be identified. In this work, many components faults and sensors faults are described in Section 7 and analyzed.

- Refrigerant leakage
  Refrigerant may leak from the system in different positions. Leakage may appear in different components or have different symptoms. For instance, the refrigerant may leak from the expansion valve, solenoid valve, or compressor. Leakage may result in compressor stop due to low suction pressure, bubbles in the sight glass, insufficient compressor lubrication, frost at the evaporator outlet, and more [7].

- Mass flow rate reduction
  The system may face a low mass flow rate if pressure drops or heat absorption exist in the suction line.

- Condenser and evaporator blockage due to fouling
  Faults such as fouling and blockage around the condenser or evaporator can be inspected by experts and are more feasible for diagnosing compared to the other faults. For example, a dirty condenser causes high compressor power consumption. The evaporator can be blocked due to the heavily frosted coil, improper positioning of goods, or refrigerant flooding. The blocked evaporator results in an increase in discharge temperature and affect system efficiency [7].

4.3 Modeling

Real refrigeration system test setup

The real refrigeration system prototype used in Papers A and B is a setup in the Bitzer Electronics laboratory. This setup consists of two parts: *Condensing unit* and *Evaporation side*. The laboratory setup, introduced in Fig. 4, includes ECOSTAR LHV7E/LHV5E condensing unit that delivers the cooling capacity to an evaporator in a cold room.

ECOSTAR is produced for supermarket refrigeration systems. Today, there are seven types of compressors by Bitzer which can be installed in the ECOSTAR unit. These compressors are, in general, 4-cylindrical and 2-cylindrical compressors. In the laboratory setup, a 4VE-7 compressor is installed, which is a 4-cylindrical semi-hermetic reciprocating compressor. The compressor's cooling capacity is defined by the percentage, which is a map of the frequency converter speed to the percentage. The compressor starts working when the frequency converter speed is 20 Hz. Therefore, the compressor cooling capacity is 0% when the converter speed is less than 20 Hz. The maximum compressor cooling capacity is 100%, which is obtained when the converter speed is 87 Hz. The other operating condition specification of the laboratory setup is given in Table 1.

Fig. 4: Condensing unit of the laboratory setup at left and the evaporation side which is inside of the cold room at right. This setup is used for data collection in Papers A and B.

Fig. 5 presents the laboratory setup components. Supermarket refrigeration systems typically consist of two parts controlled separately. Condensing unit of the laboratory setup is designed for supermarket refrigeration systems, as shown in this figure. The mentioned condensing unit has a controller called LMC300, which is parameterized according to the compressor's model to regulate compressor capacity. Data from the controller is logged and monitored using LMT software with two different modes. The first one is suction pressure control, and the second one is room temperature control. LMT controls condensing units with different hardware setups and input/output configurations.

parameters	range values
Condensing temperature (T_C)	less than 55°C
Discharge temperature (T_{dis})	less than 130°C
Saturation temperature (T_0)	more than -45°C
Pressure difference	350000 Pa
Superheat temperature (T_{sh})	less than 21°C
Compressor drive frequency	25-87 Hz
Condenser fan speed	0-100%
Cooling capacity at T_0: 5°C	30 kW
Cooling capacity at T_0: -10°C	17 kW

Table 1: Rang of operating condition for the laboratory setup using refrigerant R-134a.

Two types of inverters are used in the ECOSTAR unit: Danfoss FCM300 and Leroy Somer Varmeca 30 series. In the laboratory setup, a Danfoss FCM300 inverter operates

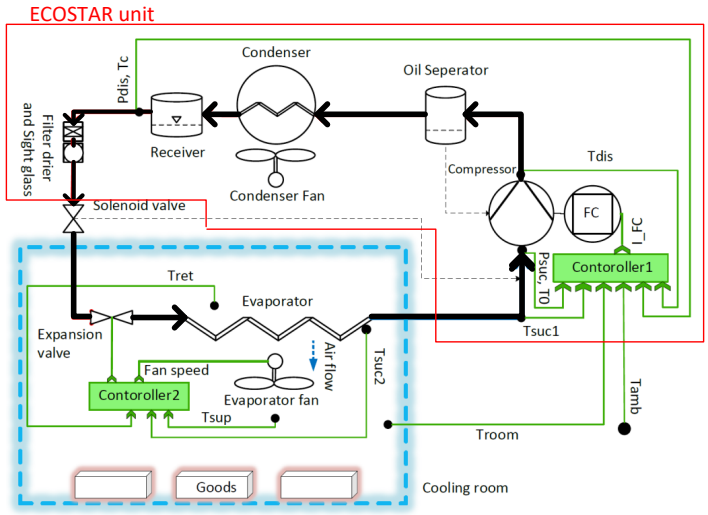

Fig. 5: Refrigeration setup used in the laboratory. Condensing unit of the test setup (ECOSTAR), which is produced for supermarket applications, is distinguished by red colour.

between 25-87 Hz. The inverter frequency depends on the cold room temperature and suction gas pressure.

A double condenser fan is installed in the laboratory setup. Each fan has a maximum power consumption of 350 W and 9070 $\frac{m^3}{h}$ air flow rate at 50 Hz.

The setup's receiver can be charged with 23.2 Kg refrigerant. There are several sensors installed in ECOSTAR units which are, in general, temperature and pressure sensors. Temperature sensors measure evaporation temperature, suction temperature, saturated temperature, discharge temperature, ambient temperature, and optional cold room temperature. The pressure sensor measures suction pressure and discharge pressure.

The evaporation side of refrigeration systems can be in different sizes and designs depending on the field's requirements. Thus, we do not specify evaporation side components and configuration. However, for an overview of the laboratory setup, an electrical expansion valve is installed before the evaporator; one evaporator and two evaporator fans are used in this setup. A controller controls superheat temperature by regulating the expansion valve opening degree and speed of the evaporator fans.

Simulation model

Several reasons encourage us to collect data from a simulation model instead of a real system. First of all, training the neural network is a process which requires data in different operation conditions. Unfortunately, the required amount of data in different operation conditions, both for non-faulty and several types of faulty conditions, is unavailable for doing experiments. In addition, data from the field cannot be trusted whether they are correctly labelled. Plus, emulating several faults in the laboratory setup and collecting data was time and cost-consuming. Thus, in Paper C, it is decided to work on the simulation model used in the company for other development purposes.

The simulation model in Matlab/Simscape is presented in Fig. 6. This model presents the main components of the refrigeration system in grey blocks. Properties of refrigerant R-134a are used through the simulation. A controller adjusts parameters such as condenser fan speed (V_{cond}), compressor drive frequency (V_{cpr}), condensing temperature (T_C), saturation temperature ((T_O), superheat temperature (T_{sh}), expansion valve opening degree (V_{exp}), and evaporator fan speed (V_{evap}). By using these parameters, the controller regulates room temperature (T_{room}), which is considered, in this model, the same as supply temperature. Heat is applied to the cold room in this model using *heat load* block. An electrical expansion valve and two-stage compressor, which run between 25-87 Hz, are selected. The model has a cooling capacity of 17 kW at ambient temperature (T_{amb})10°C and $T_{room}5°C$ [9]. The red blocks in Fig. 6 indicate faults emulation. These faults are described in Sub-sections 7.1 and 7.2 in details. The operation envelope of the simulation model can be found in Table 2.

Parameters	Value ranges
Condensing temperature (T_C)	less than 67°C
Saturation temperature (T_0)	more than -45°C
Suction pressure	less than 200000 Pa
Compressor drive frequency	25-87 Hz
Compression speed	20-100%
Condenser fan speed	0-100%
evaporator fan speed	30-100%
Cooling capacity at T_0: -10°C	17 kW

Table 2: Operating limits for the simulation model.

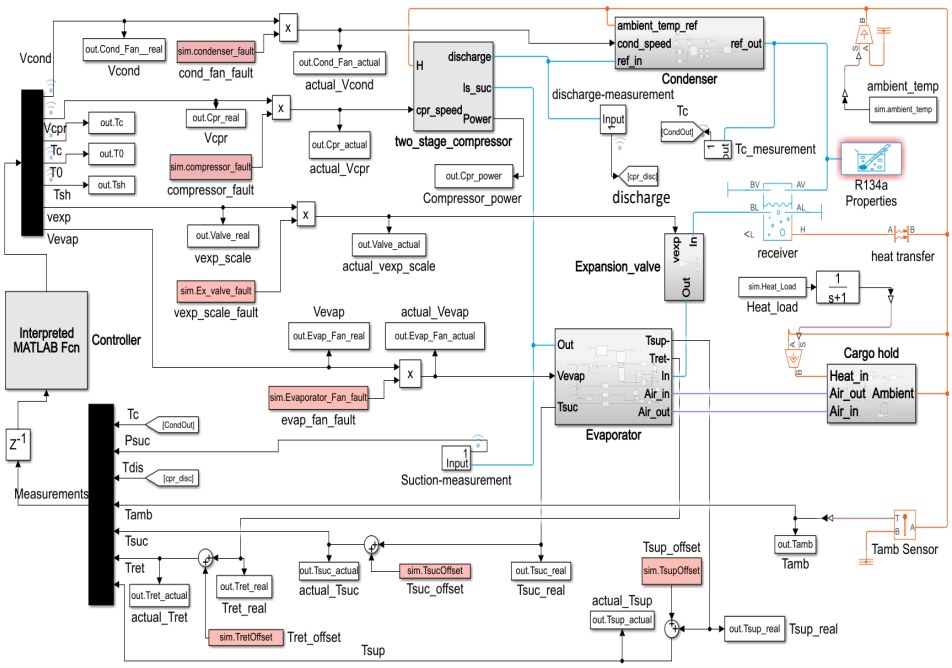

Fig. 6: High-fidelity Simscape model, which is used in Bitzer Electronics as a development prototype [9]. In this model, grey blocks indicate the main components of the refrigeration system and the red blocks are assigned to the system for fault simulation.

5 State of the art

In this chapter, first, recent fault detection and diagnosis strategies in different industries are introduced. Then, some of the recent research on fault detection and diagnosis of heating, ventilation and air conditioning systems are presented. Finally, the state-of-the-art in the field of refrigeration systems is discussed.

5.1 Fault detection strategies

Fault detection and diagnosis (FDD) is an approach to distinguishing one or more abnormal events in a system and identifying their locations. For several decades, FDD methodologies have been highlighted by both manufacturers and scientists as the development of FDD tools affects the efficiency and optimal operation of industrial systems. Moreover, the FDD tools development improves product reliability and reduces product-related costs [10]. FDD methodologies can be divided into *model-based*, and *data-driven* approaches. The model-based FDD approaches are used in different industries in which the reliability and development of FDD models rely on the precise mathematical models of the systems, see [11] and [12]. However, a mathematical model of the system is difficult to make in many industries as the operation condition, type of components and size of the system may differ. Thus, data-driven methodologies are introduced for such systems. The data-driven FDD models rely on information obtained from input/output data. Different data-driven approaches are capable in different industries depending on data complexity, correlations, non-linearity, time dependency, dynamics' frequency, etc. For instance, a combination of a Genetic algorithm with a Support Vector Machine (GA-SVM) is used in [13] for FDD purposes. Then, the faults are applied to the electromechanical actuators in electromechanical systems. [13] indicates that GA is able to improve parameter optimization in SVM and computation time. A correlative statistical analysis is combined with a sliding window algorithm in [14] for fault detection in the thermal power plant process. Subspace system identification is used for detecting and identifying actuator and sensor faults for three-Phase induction motors in [15]. In [16], a neural network-based classifier is used for leak detection. A model-based Gaussian process regression is proposed to estimate the leakage size in a water distribution network. A Fuzzy learning-based algorithm is proposed in [17] for detecting sensor faults in a converter where five types of faults: drift, bias, precision degradation, spike, and stuck, are applied to a sensor.

5.2 State of the art for FDD in HVAC systems

Heating, Ventilation, and Air Conditioning (HVAC) systems control residential and commercial buildings' expected ambient temperature, humidity, and air ventilation. HVAC systems are one of the residential areas' energy consumption sinks; Thus, improving

HVAC systems' operation and maintenance may greatly impact energy costs [18]. Several FDD approaches are investigated in HVAC applications. Artificial Neural Network (ANN)-based FDD methodologies for chiller applications are proposed in different works e.g. [19], [20], and [21]. Support Vector Data Description (SVDD) is proposed in [22] and demonstrated as an effective method for fault detection in nonlinear applications. SVDD supports functional system data in a minimum-volume hypersphere in a high-dimensional feature space. Then, faulty data falls outside the hypersphere as outliers. In [23], and [24], ML-based FDD strategies are used in air-handling applications. A statistic-based strategy is used in [18] for finding and prioritizing three common faults in HVAC systems: high cooling effort, cycle frequency, and setpoint error. [18] analyzes cloud-based data from residential buildings by comparing features between systems instead of within systems. Kernel Density Estimation (KDE) is applied to find the probability density function of each feature where the faults are considered as outliers, appearing in a low-density area. The effectiveness of the KDE method for fault detection in HVAC systems is presented in [25], in which the operational behaviour of five normal and faulty household systems are studied. [25] demonstrates the model's ability to identify malfunctioning systems among thousands of systems whose thermostats are connected to the cloud.

In [26], Deep Recurrent Neural Networks (DRNN) is proposed as a fault detection and diagnosis classifier in small HVAC systems. [26] investigates several FDD models to classify six categories: non-faulty condition, four types of valve faults in HVAC systems, outside air damper, stocked, and outside air sensor bias. [26] studies three challenges as follow:

1. Several DRNNs models with different depths and different hyperparameters are proposed.

   In order to make different DRNN structures, Input-to-memory, memory-to-memory, and memory-to-output layers of the RDNN are extended in [26] by one or more intermediate layers. Therefore, five different structures are proposed as standard DRNN, Deep Transition Input DRNN (DTI)-DRNN, Deep Transition Output DRNN (DTO)-DRNN, Stacked-DRNN, and Stacked and Deep Transition Output combined DRNN S,DTO-DRNN, see fig. 7. The author tests all the models with various Long-Short Term Memory (LSTM) units from two to four layers. The standard LSTM-based DRNN, shown in Fig. 7, is presented, which can remember the data's short and long-term temporal correlations. This methodology outperforms in FDD applications, e.g., HVAC systems better than traditional neural network models such as Back Propagation (BP) [27].

   A deeper input-to-memory function can perform as a feature extraction function in DTI- DRNN. Therefore, more information from the data can be exploited by adding the number of intermediate layers to the input layer. A deeper memory-to-output function may be more strength in the classification task. In DTO-DRNN,

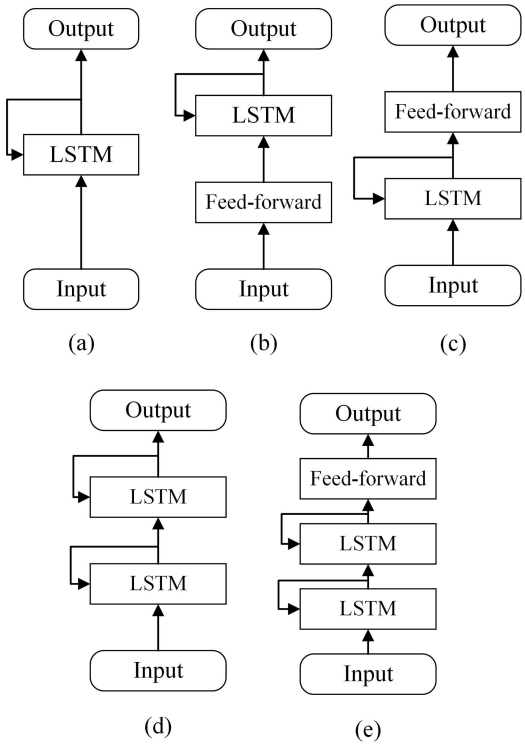

Fig. 7: Five different DRNN structures proposed in [26]. These structures are called standard DRNN (a), Deep Transition Input DRNN (DTI)-DRNN (b), Deep Transition Output DRNN (DTO)-DRNN (c), Stacked-DRNN (d), and Stacked and Deep Transition Output combined DRNN S,DTO-DRNN (e). Among these structures, a model using S,DTO-DRNN obtains the best result.

the intermediate layers are plugged into the last layer, which eases the prediction task [26]. Stacked-DRNN is introduced to stack one or more LSTM units upon each other. Then at each timescale, recurrent layers can share the features. The author in [26] proposes that one or two stacked intermediate layers may improve the FDD results compared to the standard DRNN. Another structure which may help the FDD task is S,DTO-DRNN. This structure is proposed in [26] to improve the prediction task by the DTO function; Plus, better learning can be achieved due to preserving long and short temporal correlations when using stacked-DRNN.

2. The optimal model is selected to obtain better classification accuracy and lower computation time.

   The five DRNN structures mentioned above are configured to 200 models with different numbers of LSTM and hyperparameters such as optimization problem,

drop-out, mini-batch size, layers size, etc. Among all experiments, the S,DTO-DRNN model with 2-layer LSTM obtained the best result. Ada-max is the best optimizer in this model, and a 20% drop-out could prevent model overfitting.

3. The optimal S,DTO-DRNN model is compared with an optimal Random Forest (RF) and Gradient Boosting models (GB). [26] indicates that S,DTO-DRNN model outperforms RF and GB regarding higher accuracy. Fig. 8 illustrates a comparison of S,DTO-DRNN, RF, and GB using Receiver Operating Characteristic (ROC) curves. A ROC curve represents a binary classifier performance by plotting a false positive rate versus a true positive rate in different thresholds for the prediction. Calculating Area Under the Curve (AUC) helps evaluate the classifier. Note that a higher AUC rate in the range [0,1] is preferred.

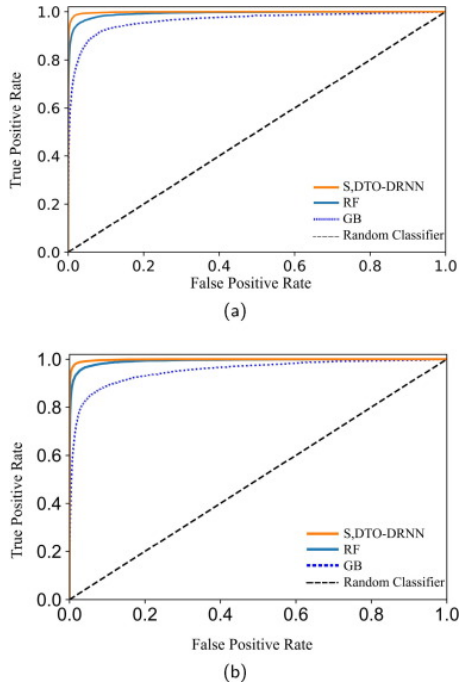

Fig. 8: Comparison of S, DTO-DRNN, RF, GB and a random classifier for classifying one of the faults from non-faulty data in [26]. Plots (a) and (b) represent the responses for two different data sets from air handling units. The curves in both plots start from zero false positives to true positives. The best AUC can be obtained by S,DTO-DRNN.

An RF classifier is combined with SVM in [28] for FDD purposes in Air handling units, see fig. 9. [28] obtained classification accuracy up to 98% and achieved a promis-

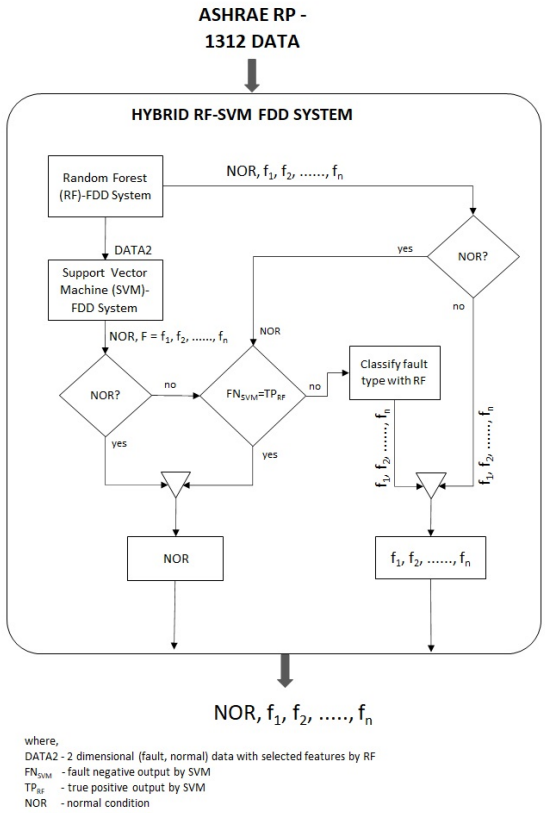

Fig. 9: The algorithm structure of the RF-SVM model used in [28]. The model classifies 14 classes with 98% classification accuracy.

ing result for building management purposes by using fewer samples for training. In [28], the model classifies 14 classes, including a non-faulty class and thirteen types of faults in an air handling unit. Fig. 10 represents the faults descriptions and the classification result using the RF-SVM model. Fig. 10 indicates no false positive obtained using this model, and several faults are classified with 100% classification accuracy. The RF-SVM model is compared with the RF and SVM models. The results show that RF and SVM could classify the data with 86% and 77% accuracy, respectively. However, both SVM and RF perform very well in the classification of some of the classes, e.g., SVM performs well for the classification of functional system condition with 100% accuracy and can be used for fault detection. [28] confirms that even though RF-SVM outperforms SVM and RF models, the RF-SVM model is computationally more expensive than the other models. [29] used another hybrid model called Extended Kalman Filter with Cost-Sensitive

Dissimilar ELM (EKF-CS-D-ELM). The same data as [28] is used in [29]. To compare the EKF-CS-D-ELM with the RF-SVM model, RF-SVM achieved a better classification accuracy. In [30], another investigation is done on the same data set as used in [28]

Fault	Abbreviation	Description
F0	NORMAL	Normal condition
F1	CCV15%OP	Cooling coil valve 15% opened
F2	CCV65%OP	Cooling coil valve 65% opened
F3	CCV100%CL	Cooling coil valve fully closed
F4	CCV100%OP	Control coil valve fully opened
F5	DLAFTSF	Duct leak after supply air fan
F6	DLBFSF	Duct leak before supply air fan
F7	EADAMPCL	Exhaust air damper closed
F8	EADAMPOP	Exhaust air damper opened
F9	OADAMP45%OP	Outside air damper 45% opened
F10	OADAMP55%OP	Outside air damper 55% opened
F11	OADAMPCL	Outside air damper closed
F12	RFFAIL	Return air fan failure
F13	RF30%SPD	Return air fan 30% fixed speed

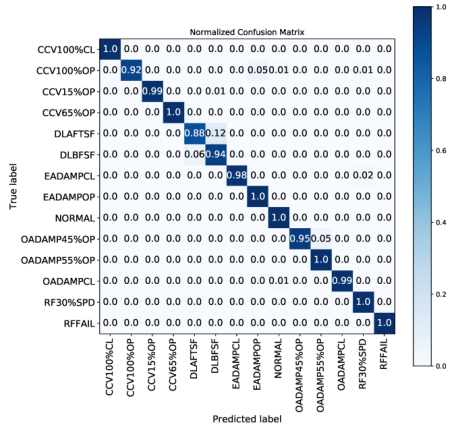

Fig. 10: The table at the top represents the faults descriptions in [28], and the result of FDD in an air handling unit using the RF-SVM model at the bottom. The diagonal values are correct classification responses showing classification accuracies in the range [88% ,100%] for all classes.

and [29] for FDD purposes. In [30], Slow Feature Analysis (SFA) is combined with feature sparse data-based dynamic to reduce data dimensions. Then, the data is infused into several FDD models, such as SVM and CNN, where better than 95% classification accuracies are achieved. Therefore, the RF-SVM model proposed in [28] outperforms the other models proposed in [30] and [29].

5.3 State of the art for FDD in refrigeration systems

Different approaches are studied regarding FDD in refrigeration systems. As refrigeration systems are the main application of this objective, We decided to investigate state-of-the-art in refrigeration systems specifically. There are both model-based and data-driven proposals for FDD in refrigeration systems. For instance, in [8], the impact of air mass flow rate reduction over each heat exchanger is explored. Thus, a dynamic FDD model for vapor-compression refrigeration systems is introduced, which is sensitive to the systems' evaporator and condenser fouling faults. Moreover, the sensitivity information is used to create a structural residual that is able to identify different types of faults. As the precise model of many systems are complex or may be unavailable, data-driven models are today more interesting for both scientists and industries [31]. There exist several investigations for FDD in refrigeration systems, e.g. [32] [33] [34]. Refrigerant leakage in four modes, namely overcharge in steady state, under charge in steady state, overcharge in the non-steady state, and under charge in the non-steady state, are considered in [34]. In this work, an ANN-based FDD model is applied to identify the faults in a laboratory setup. In [35], existing FDD in Supermarket Refrigeration Systems (SRS) are explored and categorized into four groups as follows:

- Energy-analysis-based FDD
  In this analysis, also called whole building level, the energy consumption of the HVAC&R systems in supermarkets is compared with historical energy data or standard baseline values. This method has the ability to identify a supermarket with high energy consumption even though the systems' components are fault-free.

- Thermodynamic-analysis-based FDD
  The thermodynamic behaviour of Refrigeration systems can be affected by components faults or sensor faults. Therefore, some faults can be detected by analysing the changes in temperature, pressure, mass flow rate, heat transfer rate, etc.

- Sensor-value threshold-based FDD
  In this strategy, the sensor values of refrigeration systems are compared with a pre-set threshold. This method provides information about the system's status by identifying the symptoms. However, the root cause of the symptoms is not clear. For example, In supermarket refrigeration systems, different faults can lead to unbalanced cold room temperature. The FDD model can announce that the room temperature is incorrect but can not identify the root cause of the problem.

- Alarm-analysis-based FDD
  Every feature has some limits which are already defined during the control design of the refrigeration system. In this method, an alarm raises when one or more features are distinguished out of the pre-defined limits or operation envelope. An alarm is presented as an event in the data log. Like the previous category, this

Classifiers	PCA-SVM (4 Input Vectors)		SVM (16 Input Vectors)	
	Classification Accuracy	CPU-time, s	Classification Accuracy	CPU-time, s
Fault 1	100%	1.343567	99.85%	312.9745
Fault 2	98.84%	0.041281	96.94%	9.224946
Fault 3	98.57%	0.039513	97.01%	9.498971
Fault 4	98.68%	0.017369	97.35%	5.315243
Fault 5	98.72%	0.017586	98.72%	5.265132
Fault 6	99.81%	0.005600	98.35%	1.960079
Fault 7	99.84%	0.004918	98.88%	1.494249
Fault 8	99.21%	0.005465	98.89%	0.742091

Fig. 11: A comparison between the SVM model and PCA-SVM model for diagnosing eight types of faults in Refrigeration systems regarding classification accuracy and computation time [36]. Here, PCA-SVM obtained better results than SVM model and computation time is more efficient in PCA-SVM than SVM.

method represents the malfunctioning of the system by detecting the symptoms rather than diagnosing the existing fault in refrigeration systems.

A PCA-SVM classifier is used in [36]. In this article, feature space is reduced from sixteen dimensions to four. Then, the four-dimensional data is fed into an SVM classifier to diagnose eight types of faults. Fig. 11 represents the comparison between SVM model using 16-dimensional data and PCA-SVM model using 4-dimensional data. In [36], the PCA-SVM model with a Back Propagation Neural network (BPNN) is compared. The results, introduced in Fig. 12 show that the PCA-SVM model obtains better results for FDD in vapour-compression refrigeration system than BPNN.

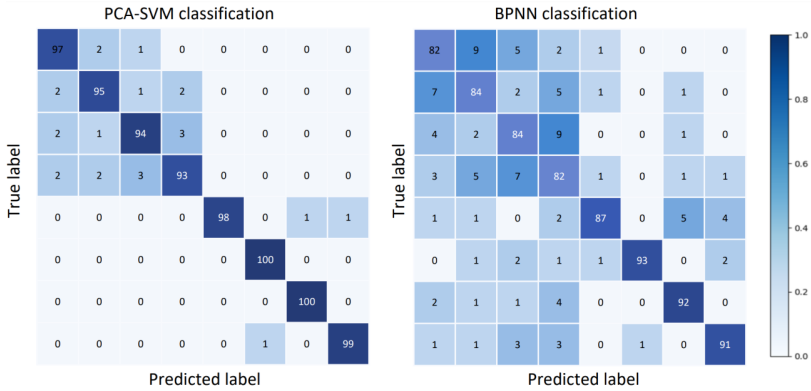

Fig. 12: According to [36], PCA-SVM outperforms BPNN model regarding diagnosing eight faults in the refrigeration system. This figure indicates PCA-SVM classification accuracy in the range [93%,100%], while BPNN obtains the accuracies in the range [82%, 93%].

6 Methods

This chapter discusses the machine learning methodologies used in this study. First, machine learning background is represented. Then, different methodologies proposed in Papers A, B, C are described. The equations and figures may be used directly or indirectly from the papers.

6.1 Machine learning Background

Machine Learning (ML) model is a predictive function that utilizes a set of examples or training data to predict, make a decision, or classify a new data set. The data is fitted to the model in the training process by adapting the parameters while minimizing the cost function. The optimized function is used to decide on unseen data. Today, data accessibility, availability of complex technologies, and easy-to-use frameworks and libraries pave the way for using ML in many industries for the development of products and processes [37]. Machine learning models can be divided into four categories when talking about the learning process:

- Supervised learning
  In this methodology, input data is labelled, and the labels are the model's true output. The model maps every input sample to the corresponding output during the training phase. This methodology replicates human learning and can be categorized into regression and classification. A supervised learning model has some pros and cons. For example, supervised learning models are usually simpler and more accurate. However, they require proper training data, which provides accurate and sufficient information. Moreover, inaccurate labelling leads to incorrect training [38].

- Unsupervised learning
  In unsupervised learning, a model extracts relations, patterns, or insights from unlabeled input data. Then, use the information for different tasks such as clustering, outliers detection or density estimation.

- Semi-supervised learning
  A semi-supervised learning model takes advantage of both supervised and unsupervised models. This model learns in the presence of both labelled and unlabelled data. Semi-supervised learning algorithms are interesting in applications in which labelling all data is either impossible or expensive [39]. Thus, the semi-supervised algorithms use unlabeled data to improve supervised learning tasks.

- Reinforcement learning
  In this category, the model learns while interacting in an environment that it receives maximum rewards accumulation and minimum penalties. Here, the agent,

who decides which action should be taken, learns from its own experience, not predefined training data set [40]. Some articles do not recognize reinforcement algorithms as a subset of ML since they do not train like supervised or unsupervised models. However, reinforcement learning is considered as a part of ML in many articles due to the ability to learn during the environment interaction [38]. The reinforcement learning method is usually used for optimization purposes. However, it is recently noticed for fault detection purposes such as [41].

This work relies on the data which is taken either from a real refrigeration system or a high-fidelity simulation model. The generated data is labelled, meaning it is more feasible and accurate to use supervised learning methods. In this study, several supervised learning models are introduced because, first, we are not sure that a system in the field does not carry one or more faults from installation or the start of data collection. Therefore, the system in a situation defined as non-faulty can be faulty in reality. Second, the supervised and offline models may be more economical as the model is trained once and can be executed several times in embedded software without retraining. Therefore, labelled data is used for fault detection, and we ensure accurate data labelling during the training process.

6.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is a supervised deep learning method which becomes more prevalent in image recognition studies and applications. The basic structure of the CNN model is a common Feed Forward Neural Network (FFNN) structure. CNN consists of artificial neurons parameterized by weights and biases. CNN model, represented in Fig. 13, is constructed from a multi-dimensional input layer, several hidden layers and an output layer. The hidden layers consist of convolutional and fully connected layers, and the output layer is fully connected. This model performs two tasks to adjust CNN model parameters during a training process. The first task is *feature extraction* which is done in convolutional layers. Then, after collecting all critical information and features of the data, the second task is *classification* of data using fully connected layers. Finally, the output layer predicts probabilities which show to what extent the data belongs to each class. Therefore, the model makes the same number of estimations as the number of classes in the output layer.

For each class, κ, input data $X_\kappa \in \mathbb{R}^{n\times c}$ is defined where n is the number of samples and c quantifies the number of features. Feature vectors are the logged features from a system which represent behaviour of the system and are defined as $\chi_j \in \mathbb{R}^{n\times 1}, \quad j = 1,\ldots,c$. CNN is a supervised learning methodology; Thus, the inputs of each class required the true outputsy_κ. $\{y_\kappa\}$ is defined as an orthonormal set of vectors in which the κ'th entry is one and others are zero. Therefore, the *classification problem* is finding a map $\mathcal{N} : \mathcal{X} \to \{y_\kappa\}, \kappa = 1,\ldots,\nu$. However, we need several examples which provide information on the system's behaviour in different operation conditions. Therefore,

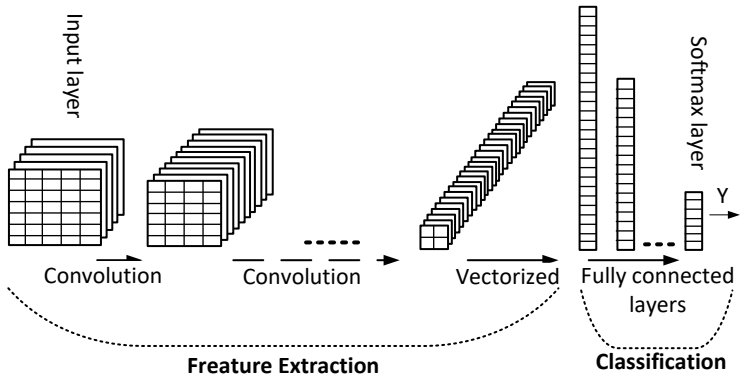

Fig. 13: A CNN model is designed to do two tasks of feature extraction and classification to predict the output. The feature extraction task is done by convolutional layers, and fully connected layers are assigned for the classification task, according to Paper A.

several data logs for each class in different operational conditions are acquired.

Data pre-processing

Collected data from a real system or a simulation model is prepared before feeding into the ML model. Data pre-processing is one of the essential and practical works in ML processes which affects the result's accuracy or computational time. Therefore, appropriate input data pre-processing is required to investigate the best model with the most satisfactory result.

CNN input may be multi-dimensional data set, mainly when it handles image recognition. In this work, the input data has two dimensions feature vectors and samples. Here, data is divided into several mini-batches by a specific number of samples in order to provide more variations of examples in the same scenarios for the training process. Fig. 14 visualizes the mini-batch process, which is done as a preprocessing step. Normalization of the data points is another preprocessing step which is done in some of the application where the variance of feature vectors are too different. The big difference between data variance causes more effect of one feature than the others. To avoid this, data normalization is suggested before the training process.

Then, all the input data from all classes are stacked upon each other as illustrated in Fig. 15. The data is ready after required preprocessing to feed into the model for training.

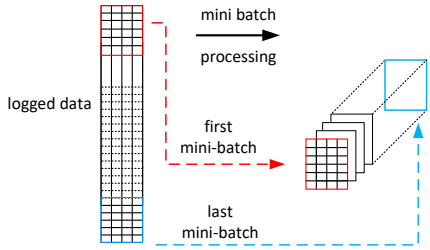

Fig. 14: Visualization of the mini-batch approach and structure of a data log after the min-batch processing. This approach rearrange our two-dimentional input to three-dimentional.

Feature extraction

As mentioned, the first task of a CNN model is extracting important features and information from the input data set. In this phase, input matrix X_κ is fed into the S number of neurons which are characterized by two parameters of weight and activation threshold (bias) matrices in an activation function as illustrated in Fig.16. The weight and bias are defined as $W^k \in \mathbb{R}^{\bar{n}\times\bar{c}}$ and $b^k \in \mathbb{R}^{\bar{n}\times\bar{c}}$, respectively..

For each κ, X_κ is partitioned into the sub-matrices $(x_\kappa)_{ij}$ for $i = 1,\ldots,n', j = 1,\ldots,c'$ where size of $(x_\kappa)_{ij}$ as the associated weights in that layer. Remark that in the partitioning of X_κ, every $(x_\kappa)_{ij}$ may have overlapping depending on the design of *Stride*. Stride is a hyper-parameter which defines the number of shifts on the elements of X_κ for partitioning. All hyper-parameters such as weights, biases, strides, etc., can be tuned to design a classifier optimally. *Pooling* is another technique which helps extract the essential features. Pooling is a filter that slides over each layer's output to do specific filtering, such as finding the maximum value in each filter or averaging each filter. Pooling is to make the output elements independent of their location in the output matrix.

The output of the last convolutional layer consists of the most important information from the input data. As it represented in Fig. 13, in order to connect the last layer to a fully connected layer, the output of the last convolutional layer should be vectorized as $Y^0 = \mathrm{col}[y_\kappa^k] \in \mathbb{R}^{n'c'S\nu\times 1}$. Y^0 would be used as an input vector to the classification phase.

Classification

Classification phase of a CNN consists of a Multi-Layer Perceptron (MLP). Typically, the last layer of MLP takes advantage of *Softmax* activation function in CNN when considering multi-class classification. For each class κ, Softmax activation function

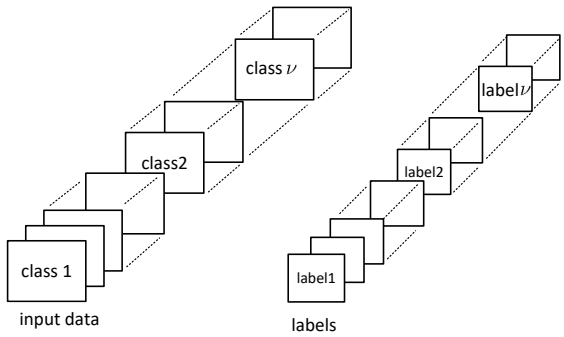

Fig. 15: Structure of pre-processed inputs and corresponding labels or true outputs. All mini-bached input data from different classess are stacked upon in a certain dimention and labels are structured in the same form. This three-dimentional stracture of the input and labels are used in Papers A and B.

converts the $\hat{Y} \in \mathbb{R}^{\nu}$ so as the κth coordinate of $\hat{Y}$ is given by:

$$\hat{y}_{\kappa} = \frac{\exp\left(Y_{\kappa}^{N_{\text{MLP}}}\right)}{\sum_{j=1}^{\nu} \exp\left(Y_{j}^{N_{\text{MLP}}}\right)} \tag{1}$$

where $Y_{\kappa}^{N_{\text{MLP}}}$ is the κth coordinate of $Y^{N_{\text{MLP}}}$ and $Y^{N_{\text{MLP}}}$ is the output of the last fully connected layer [9].

We quantify the estimation satisfaction across the training process by a loss function. The loss function can be different depending on operational applications. *Cross entropy* is one of the common loss functions After each iteration or epoch p, all the network weights W^p are adjusted to minimize the loss or error L. This adjustment is done using *Backpropagation* (BP). Thus, for each class κ a gradient of the error wrt. W^p in epoch p is presented as:

$$L_{\kappa}(\hat{y}_{\kappa}, W^p) = -y_{\kappa}^{\top} \ln(\hat{y}_{\kappa}) - (\mathbf{1} - y_{\kappa})^{\top} \ln(\mathbf{1} - \hat{y}_{\kappa}). \tag{2}$$

where $\mathbf{1} = [1, 1, \ldots, 1]^{\top}$ and $\ln(\cdot)$ is taken element-wise to yield a ν-dimensional output [42]. The gradient of the error $L_{\kappa}(\hat{y}_{\kappa}, W^p)$ is used to tune the weights until optimizing the prediction. Stochastic Gradient Decent (SGD) is an optimization approach which efficiently adjusts the weight w_i in layer l in epoch p as follow:

$$w_{il}^{(p)} = w_{il}^{(p-1)} + \alpha \nabla L\kappa(\hat{y}_{\kappa}, W^p) \tag{3}$$

where α is *learning rate* [43] [44].

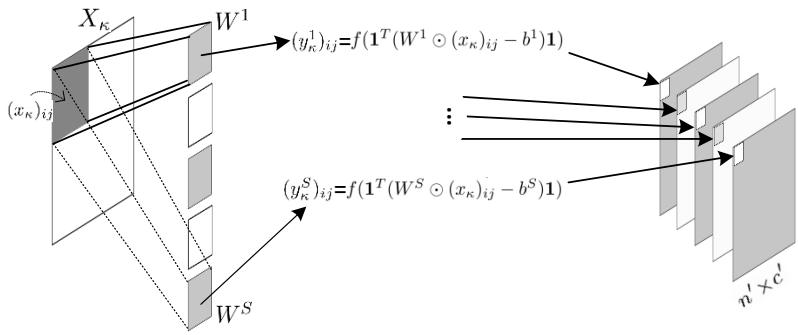

Fig. 16: Illustration of a convolutional layer which is fed by X_κ as the input and gives y_κ^k, $k = 1, \ldots, S$ as the output of the layer. $\odot$ denotes element-wise multiplication of matrices [9].

The derivative of the cross-entropy loss with respect to the weight w_i in n'th neuron is calculated by the chain rule as:

$$\frac{\partial E_\kappa}{\partial w_{n,i}} = \frac{\partial E_\kappa}{\partial \hat{y}_{\kappa,n}} \frac{\partial \hat{y}_{\kappa,n}}{\partial u_n} \frac{\partial u_n}{\partial w_{n,i}}$$

assume that in the previous layer, $u_n = \sum_i w_{n,i} x_i - b_n$ is the input to the n'th neuron [42]. The calculated loss in each epoch is propagated backwards so that optimal output estimation is obtained.

CNN has several advantages compared to other machine learning methodologies. Some of the common advantages are, first, shared weights. Comparing a fully connected neural network and CNN, CNN has considerably fewer weights as each filter has fixed values in each neuron. Thus, CNN is computationally more effective than other fully-connected models. In addition, in CNN classification, the data is location-independent when using pooling especially when it is designed deeper. Last but not least, CNN can efficiently perform anomaly detection due to the feature extraction ability, which does not feed the inessential information to the classification phase [42].

6.3 Support Vector Machines

Support Vector Machines (SVM) is a shallow learning methodology which was introduced in 1979 by Vapnik for binary classification [45] [46]. Even though SVM is a linear classification algorithm, today, this method is proposed for many non-linear classification applications due to the use of *kernel trick*. The kernel trick allows us to take advantage of SVM for non-linear classification applications, which is discussed in this section.

First, we discuss two linearly separable classes $X_\kappa,\ \kappa = 1, 2$ with labels $y_j = -1$ or $y_j = 1$, and then expand our illusion to multi-class classification and non-linear classification solutions. For each class κ, consider sample j as $x_j \in \mathbb{R}^{1\times c} for j = 1, \ldots, n$. There would be at least a linear hyperplane as

$$H = \{x \in \mathbb{R}^c \mid xw^T + b = 0\}$$

to separate classes X_κ linearly. In this hyperplane $w \in \mathbb{R}^{1\times c}$ is weight vector and $b \in \mathbb{R}$ is bias. In the training process, the hyperplane is relocated to maximize the distance between H and the nearest samples of each class. The maximum distance is called *hard margin*, and the nearest samples of each class to H are called support vectors. In other words, w and b are the solutions to an optimization problem such that $1/||w||$, which is the hard margin, is maximized. The optimization problem can be converted to:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 \tag{4a}$$

s.t.

$$y_j(x_j w^T + b) \geq 1, \quad j = 1, \ldots, n \tag{4b}$$

Even though the optimization problem can find the best values for w and b to maximize the hard margin, this solution may not be the best for classifying unseen data [47]. The hard margin may be the best solution for training, but not for verification purposes. For instance, deviation of unseen data may lead to misclassification if the hard margin is not large enough. Thus, we allow the SVM model to gently do some misclassification during the training to have a more reliable classifier when feeding unseen data [48]. The margin that some misclassification is allowed is called *Soft margin*. Fig. 17 illustrates the difference between hard margin and soft margin, in which the classifier with hard margin in the bottom has a perfect classification during the training; however, it has done misclassification during the verification phase. To find a reasonable soft margin, a slack variable ζ is applied to the optimization problems to control over the size of misclassification as:

$$\min_{w,b,\zeta} \ \frac{1}{2}\|w\|^2 + C\sum_{j=1}^{n}\zeta_j \tag{5a}$$

s.t.

$$y_j(x_j w^T + b) \geq 1 - \zeta_j \tag{5b}$$

$$\zeta_j \geq 0, \quad j = 1, \ldots, n \tag{5c}$$

where multiplier C is a hyperparameter which determines the size of determined misclassification [9] . The size of C can vary depending on the data's shape. For instance, assuming two different Input spaces in which the training data of two classes are overlapped in some area, and in the second case, The classes are separated, and there are

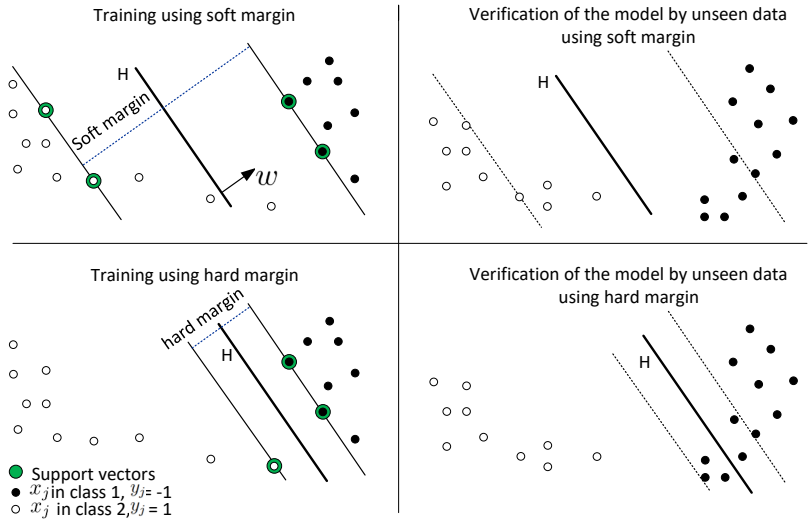

Fig. 17: The figure at the top-left presents a classifier which has a soft margin to the support vectors. The figure at the top-right shows a proper classification even though the verification data deviates. The figure on the bottom-left shows an optimized and proper classification during the training. However, the model is overfitted when verification data is used.

just some outliers. The size of C would be bigger in the first case than in the second case as the classifier must deal with overlapped samples even in unseen data in the first case. In particular, the size of C can be tuned by software aiming for high classification accuracy in the verification phase, so there is no need to choose C by trial and error.

Nonlinear classification

We already mentioned SVM as a linear classifier. However, SVM can overcome nonlinearity when linear classification is not possible. This modification can be done via a so-called *Kernel trick*. Using a kernel trick, input data is transferred from the original space to the higher dimensional space where the input data are linearly separable, see Fig. 18. The higher dimensional space in which the samples are mapped is called *feature space*. There are some common kernel functions which are used in the SVM classifier. In Paper C, Radial Basis Function (RBF) in Gaussian distribution is presented as

$$\Phi(x, x') = \exp\left(-\gamma \parallel x - x' \parallel^2\right) \tag{6}$$

where Φ is the kernel function which maps the samples to the higher dimensional feature space. x and $x' \in \mathbb{R}^{1\times c}$ are two samples that their similarity in a higher dimensional

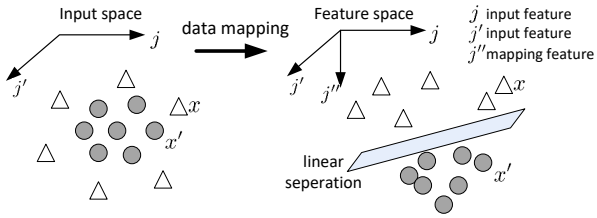

Fig. 18: The figure at left shows original data for 2 classes in 2-dimensional space, The data is mapped to the 3-dimensional space using a polynomial kernel function. At right, linear separation between classes is possible after transferring data into feature space.

space is measured by calculating Φ. Hyperparameter γ determines the influence of each sample on measuring Φ and eventually on selecting the hyperplane. d-order polynomial kernel function is described as

$$\Phi(x, x') = (xx' + t)^d \tag{7}$$

where $d \in \mathbb{N}$, $d > 0$ is the order, and $t \in \mathbb{R}$,$t > 0$ is a regulator parameter for the influence of higher and lower orders on the model.

6.4 Multi-class classification

SVM classifier is designed for binary classification problems. However, it can be used for ν-class classification where $\nu > 2$. When feeding a new sample in the multi-class classifier, SVM estimates the probability that the new sample belongs to each class. Thus, there are ν estimations as the output. One methodology is to have ν classifiers, which run in parallel to classify each class from other $\nu - 1$ classes. This method is called One versus Rest (OVR). Another methodology is to find a classifier for each class and every other $\nu - 1$ class. Therefore, there would be a total of $\frac{\nu(\nu-1)}{2}$ classifiers which run in parallel to train the SVM model.

6.5 Linear discriminant analysis

In this study, Linear discriminant analysis (LDA) is used both for dimensionality reduction and classification. LDA is described in Paper C. LDA transforms data from its original space into lower dimensional space, using transformation matrix $\Omega \in \mathbb{R}^{c \times c}$ which is defined as

$$\Omega = S_s^{-1}\ S_B \tag{8}$$

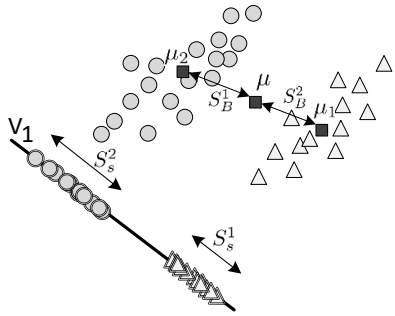

Fig. 19: An Example of 2-dimensional data that transformed to 1-dimensional data in V_1 using LDA [9].

Where $S_s \in \mathbb{R}^{c\times c}$ is a matrix in which each column is a vector of variance within each class, and $S_B \in \mathbb{R}^{c\times c}$ is a matrix in which each column indicates variance between classes see Fig. 19.

After calculating the transformation matrix, this matrix is used to generate data in the new space. However, we reduce data dimensions during the transformation to eliminate unnecessary complexity and non-linearity of the data. In this regards, eigenvalues $\lambda_i,\ i = 1, \cdots, c$ of matrix Ω are obtained and decreasingly ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \lambda'_\alpha \geq \cdots \geq \lambda_c$. If $\lambda'_\alpha \gg \lambda_{\alpha'+1}$, the most information of the data distribution is carried by the corresponding eigenvectors, which are organized in matrix V as $V = [v_1\ v_2 \cdots v'_\alpha] \in \mathbb{R}^{c\times\alpha}$. Then, data is transformed into the α'-dimensional space using matrix product $X_\kappa V$. In the matrix product $X_\kappa V$ each row is a transferred sample $r_j \in \mathbb{R}^{1\times\alpha'},\ j = 1, \ldots, n$ in class κ.

For the classification task, a neural network model is used to classify data already explained in Section 6.3. Remark that transformed samples r_j with α' feature vectors are the inputs to the classifier.

7 Experiments

This chapter includes four sections. First, different types of faults are introduced and described, which are emulated in the laboratory setup or simulated in the simulation model. Then, the data collection approaches which are done in this work are presented. Some examples of the data sets are visualized. Finally, several experiments are introduced which are not presented in the papers. These experiments were investigated during the study to improve the results of the papers.

7.1 Fault descriptions

This study considers both binary and multi-fault classifications. First, Papers A and B investigate evaporator fan fault in a real supermarket refrigeration system. Then, Paper C introduces some classifiers to diagnose twenty faults in refrigeration systems including some sensors faults and components faults in refrigeration systems. A sensor fault can be considered as a sensor drift, sensor offset, sensor freeze, sensor hard-over, and parametric fault [49]. Hard-over sensor fault describes the failure that the sensor shows some values outside of its measurable range. All of the faults investigated in this study are represented in Table 3 and labelled numerically. The labels are used for illustration of experimental results. Positive and negative offsets are applied to all available sensors in the refrigeration system model. Regarding components faults, two faults for each main component of refrigeration systems are considered. Remark that from each fault one set of data is visualized in Section 7.3. All figures in Section 7.3 are selected from the same system configuration so that readers are able to compare a specific faulty condition with another fault or with a non-faulty system. Below, every fault is described as well as the symptoms, causes, and effects of each fault.

1. T_{suc} sensor positive offset
   The suction temperature (T_{suc}) sensor is mounted in the suction line usually at the outlet of the evaporator. A positive temperature offset for 2°C is applied to T_{suc}. This sensor fault causes a larger superheat, so the system operates inefficiently. An example of a dataset when the system has T_{suc} sensor positive offset is presented in Section 7.3, Fig. 27 which is indicated as *fault 1*.

2. T_{sup} sensor positive offset
   Supply temperature (T_{sup}) is the air temperature around the evaporator fan where the heat is already transferred to the evaporator coil. A positive offset of 2°C is added to T_{sup} measurements before sending the measurements to the controller. When T_{sup} is used as a control variable, higher reading of T_{sup} causes higher superheat temperature and leads to more compressor work. An example of a data set when the system has T_{sup} sensor positive offset is presented in Section 7.3, Fig. 27 which is indicated as *fault 2*.

3. T_{ret} sensor positive offset
   Return temperature (T_{ret}) is considered the same as cold room temperature in the simulation model. T_{ret} is the temperature of the air before transferring heat to the refrigerant through the evaporator coil. A positive offset of T_{ret} causes a wrong reading of the room temperature and regulates the cold room to a lower temperature than is preferred. As a result, the failure causes an increase in the compressor's power consumption since it needs to work harder. The failure may also lead to spoilage of the goods due to the wrong reading of the room temperature. For simulating this fault, a 2 °C offset is added to the T_{ret} measurement. In Section 7.3, Fig. 27, *fault 3* represents an example of a dataset when the system has T_{ret} sensor positive offset.

4. T_{dis} sensor positive offset
   Discharge temperature (T_{dis}) is measured in the discharge line right after the compressor. In order to simulate sensor positive offset for 2 °C, the value is added to the discharge measurements before sending the measurements to the controller. This fault reduces the efficiency of refrigeration systems. An example of a dataset when the system has T_{dis} sensor positive offset is presented in Section 7.3, Fig. 27 which is indicated as *fault 4*.

5. P_{dis} sensor positive offset
   Discharge pressure (P_{dis}) positive offset is considered when the pressure sensor in the discharge line reads 10^5 Pa higher pressure than the real pressure in the discharge line. Therefore, the high-pressure switch may go off, or the compressor consumes more power than required. An example of a dataset when the system has P_{dis} sensor positive offset is visualized in Section 7.3, Fig. 27. This fault is indicated as *fault 5*.

6. P_{suc} sensor positive offset
   Suction pressure (P_{suc}) is measured in the suction line between the evaporator and compressor. Usually, the sensor is installed right before the compressor. In this study, a malfunction P_{suc} sensor with a positive offset is considered where $0.2 * 10^5$ Pa higher pressure reading than the real pressure is sent to the controller. Thus, a higher superheat temperature than requested causes extra work for the compressor and a high discharge temperature. Therefore, the system operates with low efficiency. An example of a dataset when the system has P_{suc} sensor positive offset is visualized in Section 7.3, Fig. 27 which is indicated as *fault 6*.

7. Compressor poor performance
   In reciprocating compressors, failures such as inlet and/or outlet valve loss or break, poor lubrication, pulsation of compressed gas, and dirt may lead to compressors failure [50]. These failures affect, i.e. operating speed, efficiency, refrigerant flow rate, pressure ratio, or physical damage of the compressor [51]. In this

work, the compressor's poor performance is considered when the mass flow rate is 20% less than expected. The compressor's poor performance can happen due to the loss of a valve plate in the compressor. This failure increases discharge temperature, consequently increasing compressor power consumption. An example of a dataset when the compressor has a poor performance is visualized in Section 7.3, Fig. 28. This fault is indicated as *fault 7*.

8. Loose expansion valve
   Expansion valve failure can be considered as refrigerant leakage, reduced mass flow rate, partial opening of the valve, blocked valve or loose valve. In this work, we consider a loose expansion valve where the valve opening degree is 20% more than the commanded value by the controller. This fault results in higher compressor power consumption. An example of the dataset for a loose expansion valve is visualized in Section 7.3, Fig. 28. This fault is indicated as *fault 8*.

9. Evaporator fan poor performance
   We considered an evaporator fan poorly performed when the evaporator fan speed is 20% less than the commanded value by the controller. In this fault, less ventilation is done in the cold room, and the room might have inconsistent temperatures in different spots. Furthermore, in a period, frost appears around the evaporator, and an imbalance T_{room} leads to damaged goods in the cold room. In such a system, P_{suc} is lower than the normal condition and efficiency drops [42]. An example of a dataset when an evaporator fan has a poor performance is visualized in Section 7.3, Fig. 28. This fault is indicated as *fault 9*.

10. Condenser fan poor performance
    low condenser airflow can be due to dirt or contamination. In this failure, heat can not be adequately ejected from the system to the ambient environment. Therefore, the system operates with insufficient subcooling. As a consequence, compressor power consumption increases. This fault can be simulated as the fan speed is 20% less than commanded speed by the controller which leads to lower airflow than required. An example of a dataset when the condenser fan has a poor performance is visualized in Section 7.3, Fig. 28. This fault is indicated as *fault 10*.

11. T_{suc} sensor negative offset
    A negative offset of T_{suc} sensor is chosen when the sensor shows 2 °C less than the real temperature of the suction line. Due to this failure, discharge temperature increases. Therefore, the compressor power consumption increases. An example of a dataset when T_{suc} sensor has a negative offset is visualized in Section 7.3, Fig. 29, which is indicated as *fault 11*.

12. T_{sup} sensor negative offset
    In this work, T_{sup} sensor reads 2 °C less than the real T_{sup}. Therefore, the con-

troller requests lower heat transfer than required which causes a decrease in superheat temperature and a decrease in discharge temperature. This failure may damage the compressor due to liquid slugging, which is a failure when liquid enters the cylinders of the compressor and mixes with the compressor oil. An example of a dataset when T_{sup} sensor has a negative offset is presented in Section 7.3, Fig. 29, which is indicated as *fault 12*.

13. T_{ret} sensors negative offset
A negative offset of the T_{ret} sensors is chosen by -2 °C offset in T_{ret} measurements. Therefore, insufficient heat transfer through the evaporator coil causes insufficient superheat temperature. This failure may lead to damage of the compressor. An example of a dataset when T_{ret} sensor has a negative offset is visualized in Section 7.3, Fig. 29 which is indicated as *fault 13*.

14. T_{dis} sensor negative offset
The discharge line's temperature reading drops for 2 °C to simulate the negative offset of the T_{dis} sensor. As the T_{dis} reading is lower than the real temperature in the discharge line, the compressor is forced to work harder in order to make the required temperature in the discharge line. Therefore, the power consumption of the compressor increases. An example of a dataset when T_{dis} sensor has a negative offset is visualized in Section 7.3, Fig. 29. This fault is indicated as *fault 14*.

15. P_{dis} sensor negative offset
P_{dis} sensor negative offset is simulated by decreasing the pressure measurements in the discharge line by 10^5 Pa. Due to this failure, the controller requests less heat dissipation through the condenser than required. Therefore, the condenser fan speed decreases and the superheat temperature increases. This fault leads to inefficient operation of the system because of high superheat temperature. An example of a dataset when P_{dis} sensor has a negative offset is visualized in Section 7.3, Fig. 29. This fault is indicated as *fault 15*.

16. P_{suc} sensor negative offset
P_{suc} sensor negative offset is considered when the pressure measurement in the suction line is $0.2 * 10^5$ Pa less than the real pressure of the refrigerant at the inlet of the compressor. Therefore, the compressor works less than required, and T_{dis} drops. Therefore, the higher pressure of refrigerant than expected may cause damage to the compressor. An example of a dataset when P_{suc} sensor has a negative offset is visualized in Section 7.3, Fig. 29. This fault is indicated as *fault 16*.

17. Broken compressor
Different causes may result in a broken compressor, such as wet compressor operation, system operation when T_{sh} at the evaporator outlet is too high, lack of oil,

etc. To simulate a broken compressor, we considered a compressor which passes less than 20% of the required mass flow to the discharge line. This failure can be due to damaged cylinders or valves in the compressor. An example of a dataset when the compressor is considered as broken is visualized in Section 7.3, Fig. 30. This fault is indicated as *fault 17*.

18. Blocked expansion valve
Different expansion valve failures are mentioned in fault number 8. However, for this failure, a blocked expansion valve is considered when the valve opens only 20% of the commanded opening degree. This failure can be due to a clogged or blocked valve by ice. A system with this fault operates with too low superheat. The system may also face frost at the inlet of the evaporator. An example of a dataset when the expansion valve is blocked is visualized in Section 7.3, Fig. 30. This fault is indicated as *fault 18*.

19. Broken evaporator fan
A broken evaporator fan means that one or more fans have stopped working. Therefore, heat transfer cannot be adequately done. P_{suc} and T_{sh} may be affected and T_{sh} increased. Consequently, the system operates with low efficiency, high T_{dis}, frost around the evaporator coil, and room temperature drifts from the set point [42]. In this work, we consider a broken fan when its speed is less than 20% of the commanded value by the controller. An example of a dataset when the evaporator fan is broken is visualized in Section 7.3, Fig. 30. This fault is indicated as *fault 17*.

20. Blocked condenser fan
A condenser fan works poorly or totally stops for different reasons, such as mechanical or electrical problems of the condenser fan or when the fan is blocked by dirt. In this work, we consider a blocked condenser fan whose speed is less than 20% of the commanded value. Too high condensing pressure may cause switching off due to the high-pressure limit being violated. In addition, due to low coolant, compressor power consumption increases. An example of a dataset when the condenser fan is blocked is presented in Section 7.3, Fig. 30. This fault is indicated as *fault 18*.

Table 3: simulated faults and their labels used in Paper C.

Label	Fault
1	T_{suc} sensors positive offset
2	T_{sup} sensors positive offset
3	T_{ret} sensors positive offset
4	T_{dis} sensors positive offset
5	P_{dis} sensor positive offset
6	P_{suc} sensor positive offset
7	Compressor poor performance
8	Losse expansion valve
9	Evaporator fan poor performance
10	Condenser fan poor performance
11	T_{suc} sensors negative offset
12	T_{sup} sensors negative offset
13	T_{ret} sensors negative offset
14	T_{dis} sensors negative offset
15	P_{dis} sensor negative offset
16	P_{suc} sensor negative offset
17	Broken compressor
18	Blocked expansion valve
19	Broken evaporator fan
20	Blocked condenser fan

7.2 Data collection

In this study, data is collected from a real refrigeration system and high-fidelity simulation model of the refrigeration systems in use in Bitzer Electronics. In this section, the evaporator fan fault, which is emulated in the laboratory setup, is first described. Then, the simulations of faults in the Simscape model are illustrated.

Real data collection

In the laboratory, a real supermarket condensing unit produced by Bitzer is connected to a cold room; see Fig. 4. The cold room evaporator has its own controller that controls the expansion valve opening degree, superheat temperature and evaporator fan speed. The condensing unit does not have access to the information of the cold room controller. The laboratory evaporator has two evaporator fans. One of the fans is manually switched off and is not controlled by the controller to emulate the fault scenario on the system, as shown in Fig. 20. We emulate this fault in Papers A and B as a scenario when one of the fans is broken. In order to gather data at a specific temperature and running mode

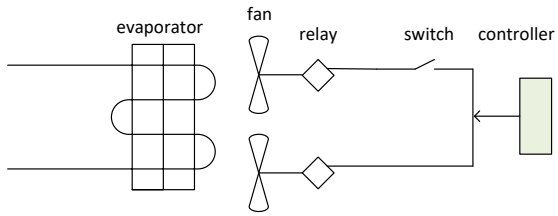

Fig. 20: Emulation of evaporator fan fault on the laboratory setup for data collection purposes in Papers A and B. Here a switch is placed such that one of the fans can be forced to stop, and the data is collected when the system has lower air ventilation around the evaporator.

of the compressor, a heat load is set in the cold room. We can collect data on specified compressor speeds by adjusting the heat load to certain values. In papers A and B, data from the laboratory setup is taken in a full-speed operating compressor when the heat load in the cold room is 17 kW. Moreover, the data is taken in the minimum speed operating compressor when the heat load in the cold room is 3 kW. In order to provide sufficient information for the machine learning model's training, data must be taken from the system in different operational conditions. In this regard, data covers system operational conditions when the heat load is in the range [3,17] kW, the setpoint varies in the range [0,12]°C, and T_{amb} varies in the range [15-27]°C. An example of data collected in the laboratory is visualized in Fig. 20 in Section 7.3. This figure illustrates that early fault detection of evaporator fan prevents noticeable consequences such as ice accumulation around evaporator, more frequent defrosting of the system and more power consumption due to higher compressor speed.

Artificial data collection

The Simscape model is already presented in Fig. 6. This part illustrates the simulation of faulty scenarios in the Simscape model. Data is collected in different system operational conditions to ensure that the data provide enough information during the training phase of the machine learning classifier. First, data is collected with the variation of heat load in the range [3,18] kW, and the setpoint in the range [0,12]°C. This data set has 12 dimensions, namely P_{dis}, P_{suc}, T_0, T_C, T_{dis}, T_{ret}, T_{suc}, T_{sh}, T_{sup}, P_{cpr}, V_{cond}, ρ. Moreover, Paper C indicates the importance of applying a variation of ambient temperature to the training data, which results in better model performance when feeding unseen data in the verification phase. Therefore, a new set of training data is collected in which data is excited by variation of T_{amb} in the range [10,30]°C. In addition, ambient temperature and setpoint temperature are added to the features of the input data while recording the data. Therefore, the recorded data set with more excitation has 14

dimensions. Some examples of logged data can be found in Section 7.3.

- T_{suc} sensor offset
  Fig. 21 illustrates the evaporator block in the Simscape model. As it appears in this figure, there is an output called T_{suc}, which is the sensor measurement. In order to apply T_{suc} sensor offset, the offset needs to be added to the sensor measurements, which can be found in Fig. 6. Moreover, real T_{suc} and other fluid characteristics of the refrigerant are sent to the compressor by block *out*, as the compressor behaves based on the real refrigerant characteristics which pass through it. Both positive and negative offsets of T_{suc} sensor are gathered when -2 and 2 °C are applied to the block called *"Tsuc-real"*, respectively.

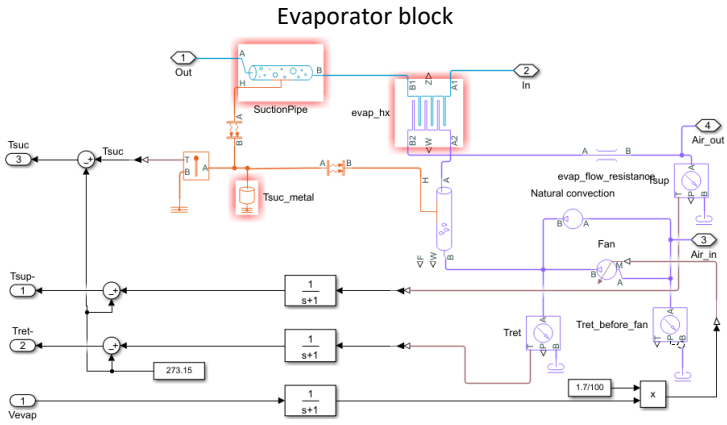

Fig. 21: The Simscape model of evaporator used in this study, for data collection.

- T_{ret} sensor offset
  In the simulation model, T_{ret} is the same as room temperature and setpoint. Therefore, any offset in T_{ret} sensor affects the room temperature. For example, suppose the T_{ret} sensor has a 2 °positive offset. In that case, it means that the compressor works harder than required to regulate the room temperature for 2 °C colder than the set point temperature. the T_{ret} measurement is calculated in the evaporator block shown in Fig. 21. Then as it can be seen in Fig. 6, the 2 °C positive and -2 °C negative offsets of T_{sup} sensor are added using the red block called *"Tsup-offset"*.

- T_{sup} sensor offset

Positive and negative offsets of the T_{sup} sensors are applied to the Simscape model in the same way as T_{sup} and T_{suc} offset, see Fig. 6.

- T_{dis} sensor offset
  T_{dis} is measured from the discharge link in Fig. 6 where temperature measurement is calculated from thermodynamic characteristics of refrigerant as it is shown in Fig. 22. 1 °C positive and negative offsets are applied to the T_{dis} measurements as appears in Fig. 22.

- P_{dis} sensor offset
  The Simscape model uses refrigerant pipe links where it is not possible just to offset the pressure because it would break the equations. Therefore, the positive and negative 100000 Pa offsets are applied to the P_{dis} measurement. P_{dis} measurement is obtained from refrigerant characteristics in the discharge link; see Fig. 22.

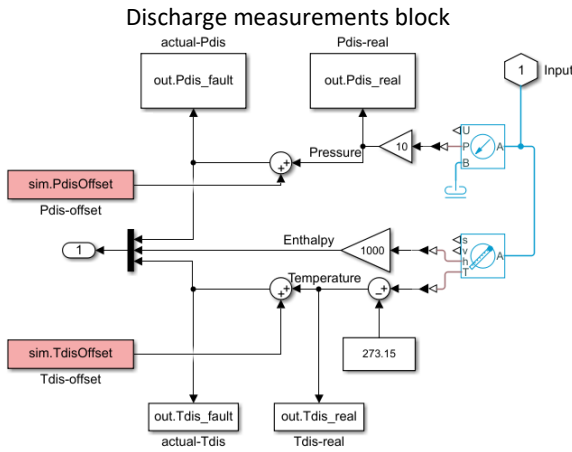

Fig. 22: The detail of *"discharge-measurement"* block in Fig. 6 including P_{dis} and T_{dis} measurements.

- P_{suc} sensor offset
  P_{suc} measurement is calculated based on the refrigerant characteristics in the suction link, which is indicated by the block called *"suction-measurement"* in Fig. 6. Positive and negative 20000 Pa offset is applied to the P_{suc} measurement as shown in Fig. 23.

- Compressor faults
  The compressor in the Simscape model is a volumetric flow device, which accu-

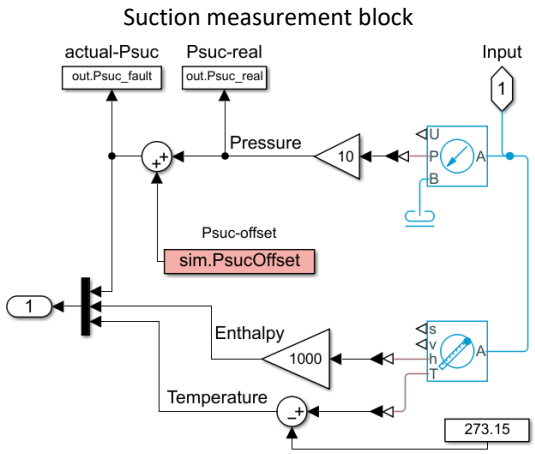

Fig. 23: The detail of *"suction-measurement"* block in Fig. 6, in which P_{suc} offset is applied to the P_{suc} measurements.

rately reflects a reciprocating compressor's flow characteristics. This study applies two faults to the compressor: poor performance and a broken compressor. Fig. 6 introduces the compressor fault simulation on the Simscape model where V_{cpr} is the controller command to the compressor. For applying the fault scenarios to the compressor, the block called *compressor-fault* affects the controller command for V_{cpr}. Thus, the compressor speed is 20% or 80% less than the required V_{cpr} commanded by the controller. These faults are called poor performance of compressors and broken compressors, respectively.

- Expansion valve faults
  The opening degree of the expansion valve is controlled based on regulating T_{sh} to maintain 10 °K. In this study, the expansion valve faults are defined as a loose expansion valve when the valve is open 20% more than commanded opening degree by the controller and blocked expansion valve when the valve is open 20% less than commanded value by the controller. To simulate these faults, as shown in Fig. 6, the controller command is affected before sending it to the expansion valve.

- Evaporator fan faults
  The controller controls the evaporator fan speed to regulate the amount of heat transfer in the evaporation phase. To simulate evaporator fan fault, the block called *"vexp-scale-fault"* affects the controller command for V_{exp} before applying the command in the fan's block diagram, see Fig. 6. In Paper C, two faults

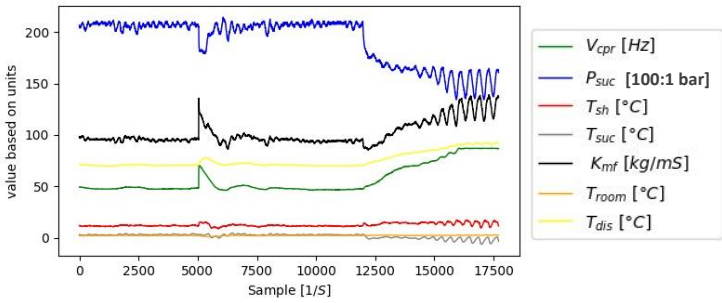

Fig. 24: An example of the data log, taken from the laboratory setup. In this example fault occurs at sample 5000. The controller could compensate the error to some extent after a while. However, existing error in the system for a longer period causes remarkable changes in all measurements. Papers A and B attempt to detect the fault before these big changes affect the system. Here, k_{mf} is a measured proportional mass flow rate at the inlet of the compressor which is independent to the compressor physical characteristics such as the inlet area.

are applied to the evaporator fan. Poor performance of the evaporator fan and a broken evaporator fan are defined where the evaporator fan speed is 20% and 80% less than the controller command, respectively.

- Condenser fan faults
  Condenser fan fault is one of the common faults in the field. The heat transfer between the condenser coil and the ambient environment can be lowered for several reasons, such as existing dirt around the condenser, partially or completely blocking the condenser fan, broken condenser fan, etc. In this study, we simulate two condenser fan faults. Poor performance of condenser fan is simulated where the fan operates with 80% of the controller command for V_{cond}, and a broken condenser fan is simulated where the fan operates with only 20% of the controller command for V_{cond}. The faults are applied by the block called *"cond-fan-fault"* in Fig. 6. Then, the affected command, which is less than the required fan speed, is sent to the condenser fan.

7.3 Data visualization

This section brings some examples of data sets collected during this study. First, an example of data collected for Papers A and B is introduced. Then, One example for each class, which is simulated and used in Paper C, is visualized.

Fig. 25: The effect of evaporator fan fault on the evaporator of the laboratory setup at sample 17700 in Fig. 24. Papers A and B aim to detect the evaporator fan fault in the system before ice accumulates around the evaporator coil or cold room temperature becomes uneven. In this experiment defrosting mode of the system was off.

In papers A and B, the evaporator fan fault was emulated on the laboratory setup. In order to give more information to the training algorithm, six scenarios with variations of heat load and setpoint are made. Therefore, six data sets are used for the non-faulty condition and six data sets for the evaporator fan fault condition. Fig. 24 is one of the data sets taken from the laboratory setup, in which the system runs functionally up to sample 5000. Afterwards, one of the fans is switched off, and the rest of the samples represent data when the evaporator fan is broken/blocked.

In the setup which is used in this study, the fault does not affect the system operation, so the fault can be detected immediately. The effect of the fault appears in the data around sample 12000 in Fig. 24 even though the fault happens at sample 5000. The effect of the fault can be visible to the owner after the operation of the faulty system for some hours or even a day when ice is accumulated around the evaporator, see Fig. 25. In the case of defrosting, the owner would experience higher than normal defrosting frequency.

In Paper C, 360 data logs are used where for each fault 18 scenarios are defined. These scenarios are defined such that the training data for each class covers variations of ambient temperature, variations of heat load in the cold room, and variations of setpoint. Heat load variations are selected such that data from different compressor speeds are achieved. In this part, one data log for each fault is introduced to visualise how the system reacts differently or closely for different faults. All the graphs in this part represent the refrigeration system Simscape model response in the same operating condition of 30 °C for T_{amb}, 7 °C for the setpoint and 8 kW for the heat load. For each data log, first, the system operates functionally up to sample 6000, see Fig. 26. Then, each fault is applied to the simulation model. Therefore, Non-faulty data is collected to sample 6000, and faulty data is collected from samples 6001 to 12000. Fig. 27 represents

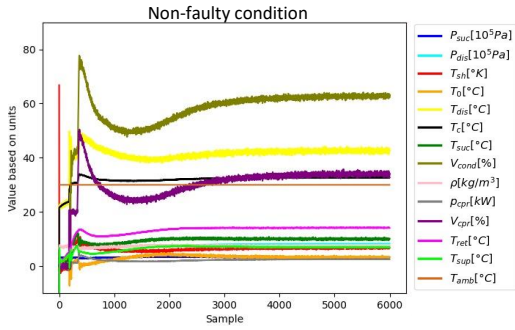

Fig. 26: An example of data log for non-faulty condition where T_{amb} is 30 °C, the setpoint is 7 °C, and the heat load is 8 kW.

positive offset of the sensor, which are indicated as faults 1 to 6 in Table 3. Fig. 28 shows an example of data for faults 6 to 10 in Table 3. Fig. 29 represents data for negative offset sensors, which are indicated as faults 11 to 16 in Table 3, and Fig. 30 illustrates faults 17 to 20 in Table 3.

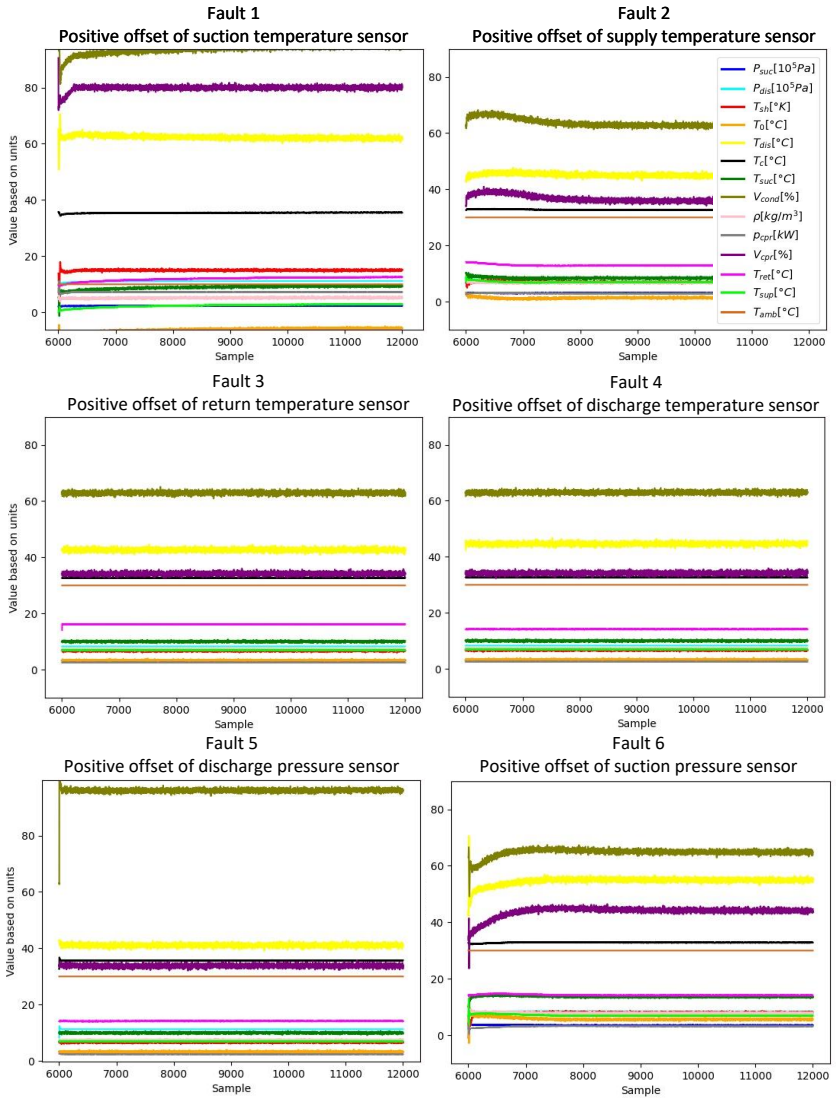

Fig. 27: Some examples of data logs for Positive offset of the sensors where T_{amb} is 30 °C, the setpoint is 7 °C, and the heat load is 8 kW.

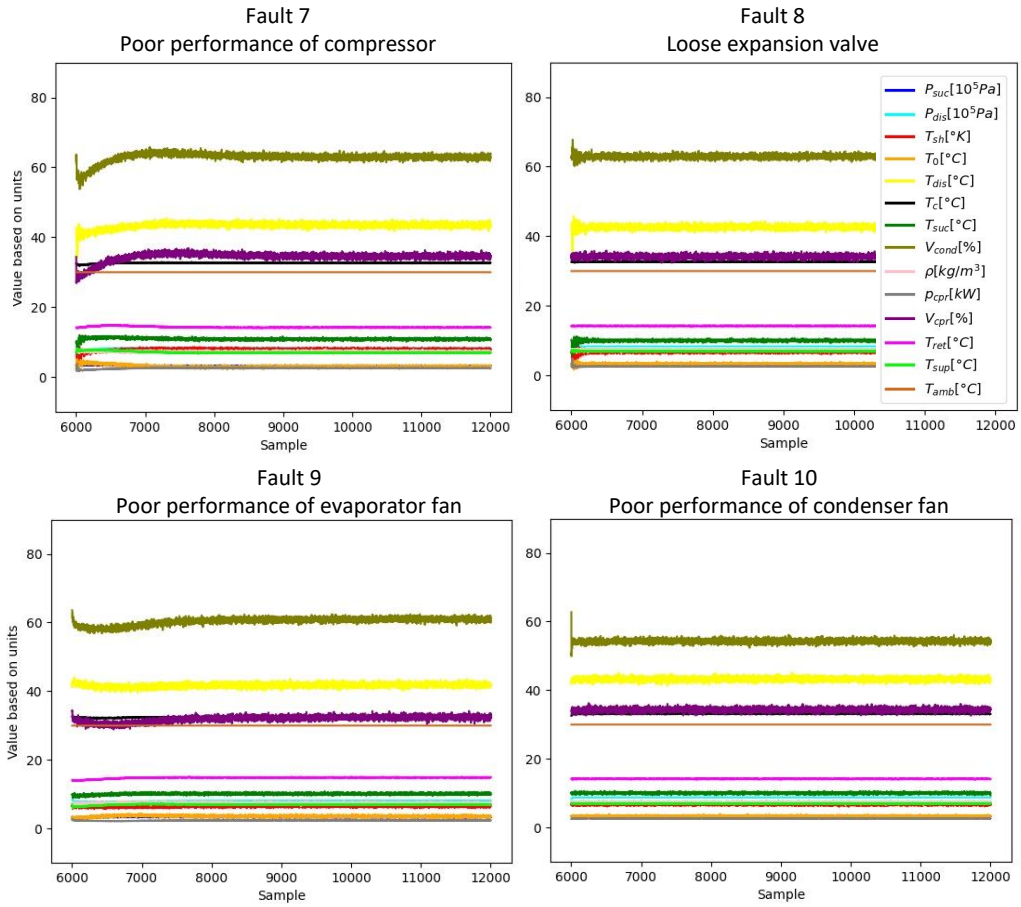

Fig. 28: Some examples of data logs for the components faults such as compressor poor performance, loose expansion valve, evaporator fan poor performance, and condenser fan poor performance. ´For all these examples, T_{Tamb} is 30 °C, the setpoint is 7 °C, and the heat load is 8 kW.

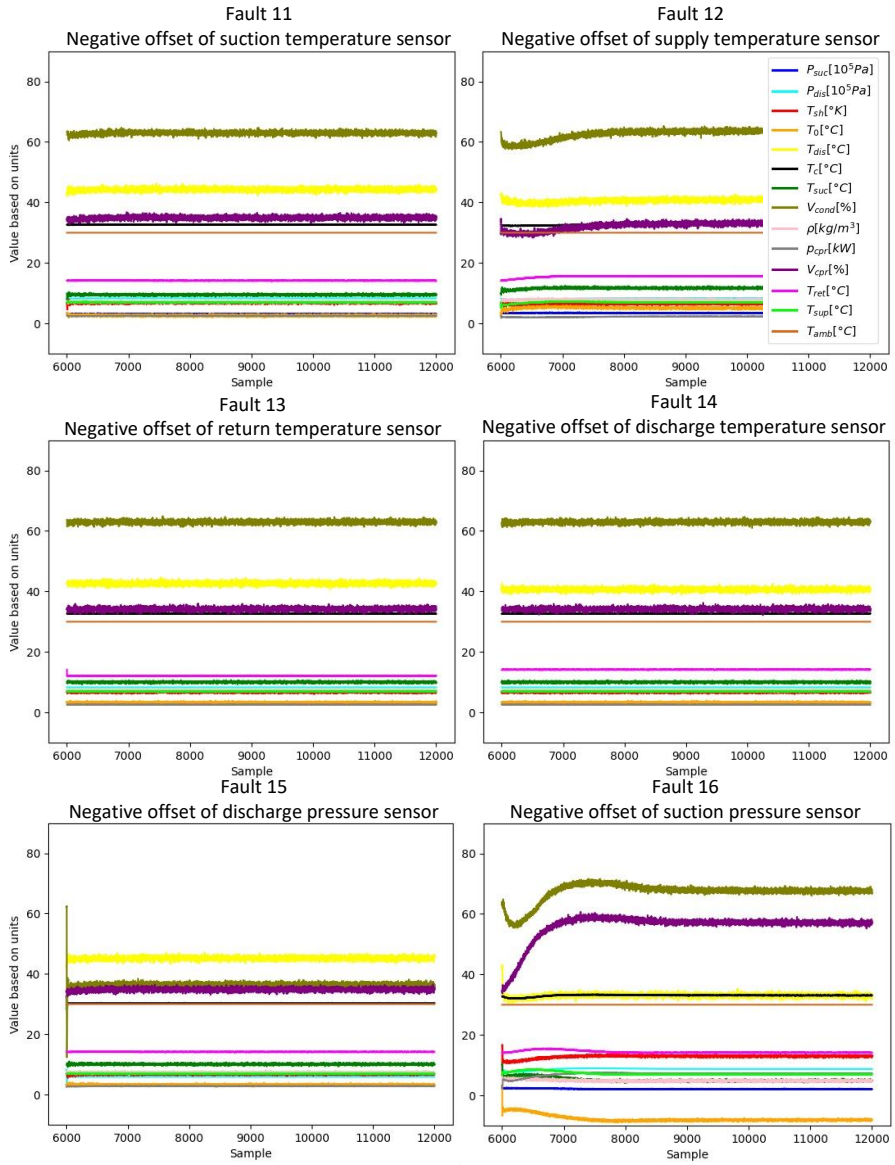

Fig. 29: Some examples of data logs for negative offsets of the sensors where T_{amb} is 30 °C, the setpoint is 7 °C, and the heat load is 8 kW.

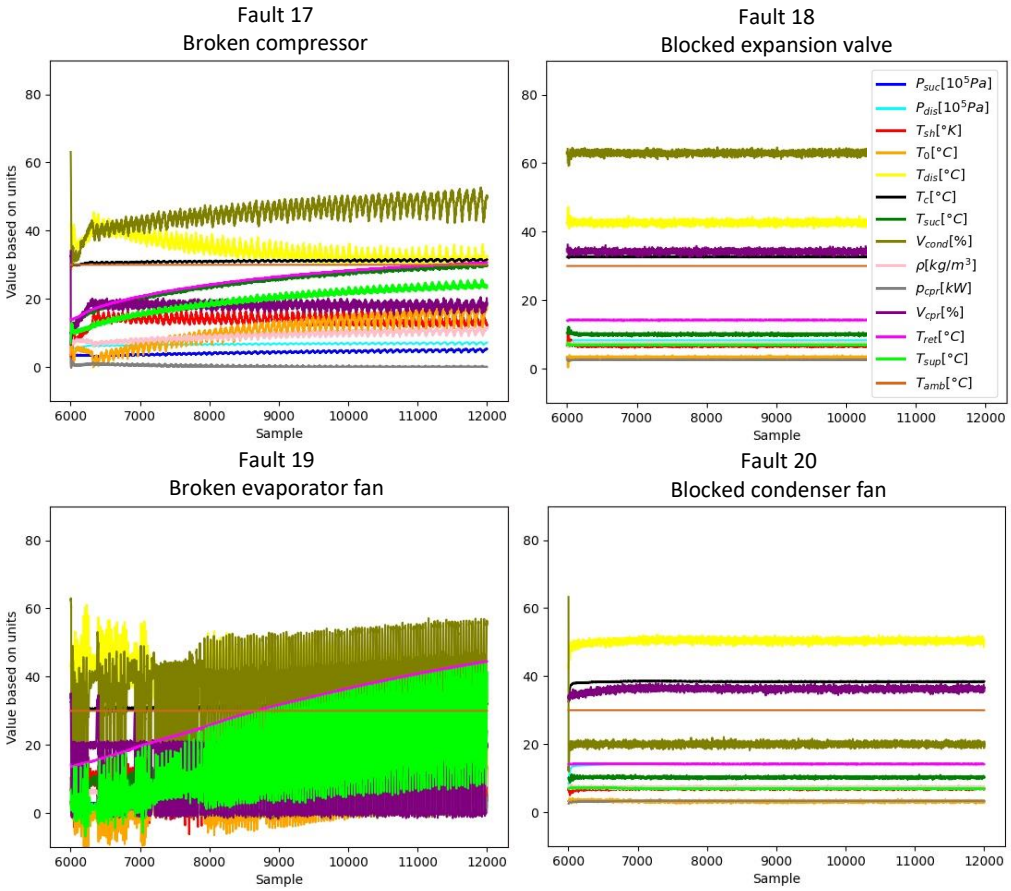

Fig. 30: Some examples of data logs for the components faults such as broken compressor , blocked expansion valve, broken evaporator fan, and blocked condenser fan.´For all these examples, T_{amb} is 30 °C, the setpoint is 7 °C, and the heat load is 8 kW.

7.4 Experiments

During this study, several experiments are done which are not presented in the papers. These experiments are described here. These experiments may lead to the paper's results or illustrate the final results in detail.

Experiment 1

In Paper A, CNN is used for fault detection and performs binary classification between Non-faulty conditions and evaporator fan malfunctioning. Different hyperparameters are tested to obtain a model with satisfactory classification accuracy and a false positive rate. Table 4 illustrates an experiment in which the CNN algorithms with different activation functions mentioned in Table 4 are evaluated. The input data used in this experiment is the real refrigeration system data used in Paper A. Note that in this experiment, the same activation function is used for all the layers except the last layer. The last layer uses a Sigmoid activation function in all tests as it can deal with binary classification in the output layer. The results in Table 4 indicate that ReLU, ELU, and Tanh activation functions perform very close to each other in which the accuracy is high, false positive rates, and losses are low. However, the Sigmoid and especially Identity activation functions do not perform as well as ReLU, ELU, and Tanh activation functions.

Table 4: An investigation on using different activation functions in the CNN algorithm used in Paper A. This table shows a better result obtained by using ReLU activation function considering the classification accuracy, value of loss and false positive rate.

Activation function	accuracy	loss	false positive
ReLU	98%	0.065	1%
ELU	97%	0.07	1.7%
Tanh	97%	0.065	1.4%
Sigmoid	96%	0.1	2.8%
Identity	93%	0.11	5.5%

Experiment 2

In paper C, a CNN classifier is used to classify 21 classes, including 20 types of faults and non-faulty conditions. The input data to this classifier is 12-dimensional simulated data. The CNN algorithm used in Paper C is specified in Table 5. In this experiment, Table 6 illustrates some performance tests using different activation functions for the hidden layers of the CNN. Here, the output layer of each model takes advantage of the Softmax activation function, described in 1 to deal with multi-class classification tasks. The results in Table 6 indicate a narrow difference among non-linear activation

Table 5: Design of the CNN algorithm, used in Paper C. Here, Sf stands for the size of the filter,Nf for the number of filters, f for the activation function and MP for the size of maxpooling.

layer	Sf	Nf	f	MP
convolution	(2,10)	16	ReLU	(1,3)
convolution	(2,3)	32	ReLU	-
convolution	(1,3)	64	ReLU	-
Flatten	-	-	-	-
F-C	300	-	ReLU	-
F-C	21	-	softmax	-

Table 6: Performance comparison of the CNN algorithm in Paper C when using different activation functions. Here. ReLU activation functions seems the best choice looking at the classification accuracy. However, using Sigmoid activation function, the CNN model can classify the non-faulty condition better than the other models.

Activation function	accuracy	loss	false positive
ReLU	94%	0.01	58%
ELU	93%	0.01	51%
Tanh	90%	0.03	49%
Sigmoid	93%	0.01	47%
Identity	91%	0.03	100%

functions. In this exercise, ReLU has the best classification accuracy and the Identity activation function which is a linear function, has the lowest accuracy, and the false positive rate shows that the classifier cannot detect non-faulty data.

Experiment 3

In Paper B, data from a condensing unit of a real supermarket refrigeration system is collected in two conditions. First, data is acquired in a functional condition called *non-faulty data*; then, data is taken from the same system with a broken evaporator fan. SVM classifier is proposed to detect the faulty condition from the non-faulty condition. In this example, the same data as paper B is selected to illustrate the effect of hyperparameters in the SVM classifier, such as selection parameters C, γ, and d where the model uses the Polynomial kernel function. In this regard, Principle Component Analysis (PCA) is used for visualization, where we reduce data dimensions from 14 to 2. PCA is described in Paper B. In Fig. 31, the PCA-SVM classifier is presented using a polynomial kernel function with a different degree. As it appears in this figure, a trade-off between hyperparameter C and d is required to find the best fit for the classifier

so that the model is not overfitted to the training data. Increasing C and increasing d at the same time may not necessarily change the performance of the classifier; for instance, the algorithm used $C = 0.01$ and $d = 3$ classifies data with almost the same behaviour as the one with $C = 1$ and $d = 7$. The classification accuracy can be found in Table 7, showing that increasing the polynomial degree does not help the classification accuracy in this case. It seems that the non-faulty condition data has more variance than the faulty condition. Therefore, hyperparameter C should be selected so that it gives enough space for variation of non-faulty samples and, at the same time, takes care of the correct classification of the faulty samples. Thus, selection of C= 1 or C =10 where d =3 are the best choice in this experiment.

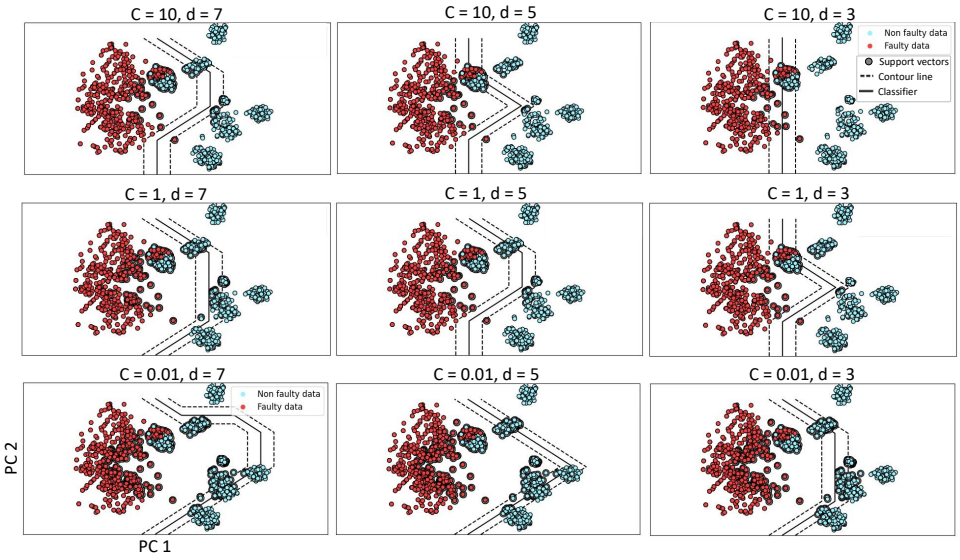

Fig. 31: PCA-SVM results using polynomial kernel function where the polynomial offset is 1, with different polynomial order from left to right and different parameter C from top to bottom. Changing the parameter C and polynomial order would affect the selection of support vectors and the shape of the classifier and contour lines. In this example, decreasing parameter C leads to a lot of misclassification in the non-faulty set. In addition, increasing the polynomial order does not necessarily improve the classification.

Experiment 4

In Fig. 32, an RBF kernel function with different hyperparameters, C and γ is tested for the PCA-SVM algorithm in paper B. From this figure, a trade-off between γ and C

Table 7: Comparison of the PCA-SVM model where polynomial kernel function is tested with different C and d parameters. Here, the best result obtined using d=3, and the higher polynomial degrees end up to the less accurate classification results.

polynomial	
hyperparameters	accuracy(%)
C: 10 , d: 7	92.3
C: 10 , d: 5	92.6
C: 10 , d: 3	95.6
C: 1 , d: 7	91.8
C: 1 , d: 5	92.6
C: 1 , d: 3	97.1
C: 0.01 , d: 7	80.3
C: 0.01 , d: 5	81.8
C: 0.01, d: 3	90.8

is required so that first, the classifier does not unnecessarily tide up the non-faulty data set which has more variance by careful selection of γ, and second, The classifier deals with the outliers properly by careful selection of C. In this case, $\gamma = 1$ and $C = 1000$ or $\gamma = 1$ and $C = 10$ seem to be the best choices. Using the classifier with $\gamma = 1$ and $C = 10000$ is not suggested, though its classification accuracy is high in Table 8; because the classifier restricts the faulty data variation in unnecessary spots. Table 8 presents the classification accuracy when the hyper-parameters such as the kernel function, parameters C, and γ are different in the algorithm. In addition, several tests are done using an identity kernel function. In Fig. 32, linear classification using $C = 10$, $\gamma = 0.01$ seems to be feasible for 2-dimensional data. However, the results of the classification using RBF are better than the linear classifier in Table 8. Comparing Tables 7 and 8, classification accuracy is higher when using the RBF kernel function. Remark that too big and too small values for C and γ are unreliable as they lead to overfitting or underfitting of the model [44]. However, they are used in this example to visualize the effect of these parameters as more tangible. The data used in this experiment is collected from a real system in the laboratory and PCA is used for dimensionality reduction to visualize the experiment in 2-dimensional space.

Experiment 5

The PCA-SVM classifier is analyzed in Paper B, where the model is verified using various verification tests, e.g. data resolution, noisy data, perturbed data, and data with disturbances as follows:

- Sample rate
  This experiment acquired data from a real refrigeration system in the Bitzer elec-

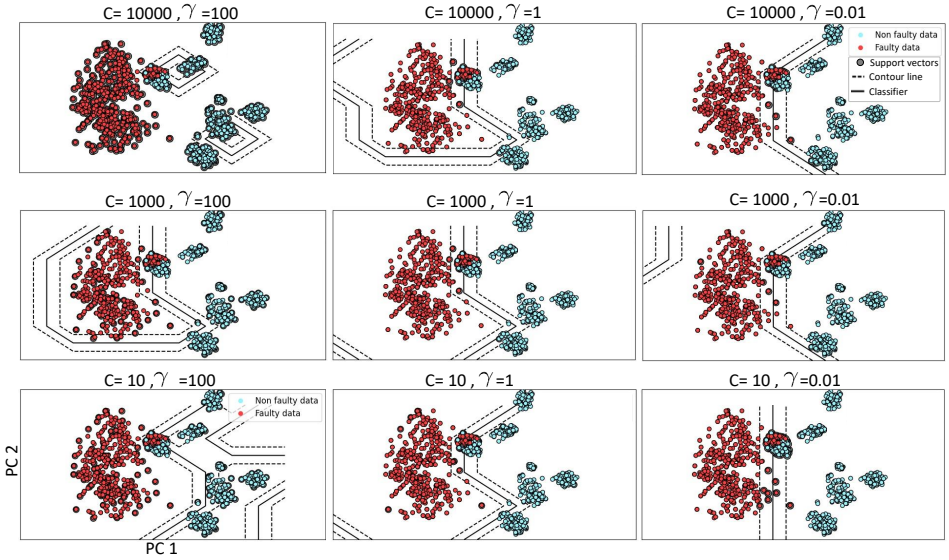

Fig. 32: PCA-SVM responses are depicted in 9 figures in which hyperparameter C is decreased from top to bottom and parameter γ is increased from right to left. Changes in parameters C and γ affect the shape of the classifier and contour line as well as a selection of support vectors. For example, the classifier selects most of the samples as support vectors when choosing the highest value for C= 10000 and γ=100. This model is not appropriate due to overfitting the model. The selection of these hyperparameters is tested to find a trade-off between them so that we have a model which has high accuracy when feeding unseen data.

tronics laboratory. This data has a 1 Hz sample rate initially. However, data logs from different systems in the field have various sample rates. Therefore, it is essential to investigate the robustness of the classifier when a trained classifier is used to deal with data from systems with different sample rates. Thus, data samples are selected as 1 sample per 1s (1 Hz),1 sample per 10 s (0.1 Hz), and 1 sample per 60s (0.01 Hz). Fig. 33 introduces the results of several experiments in which data with different sample rates and lengths are used for training. The results indicate these sample rates do not affect the classification accuracy as the dynamic of the refrigerant through refrigeration systems is lower than those tested sample rates. Training time will be longer when more samples are used. However, the test with the data with a length of 900 samples achieved better accuracy compared to those tests with fewer samples, as indicated in Fig. 33. Therefore, we can consider it the best length of data for training this classifier.

Table 8: Accuracy comparison for PCA-SVM algorithm using different hyperparameters. Here, different hyperparameters are investigated using either RBF or linear kernel function. In general RBF works better than linear kernal function. A trade off between C and γ is required to achieve the best result.

RBF		Linear	
hyperparameters	accuracy(%)	hyperparameter	accuracy(%)
C:10000 , γ: 100	96.5	C: 10000	95.1
C:10000 , γ: 1	98.3	C: 1000	95.3
C:10000 , γ: 0.01	96.6	C: 100	95.6
C:1000 , γ: 100	97.6	C: 10	95.8
C:1000 , γ: 1	98.3	C: 1	95.5
C:1000 , γ: 0.01	96	C: 0.1	0.933
C:10 , γ: 100	96.3	C: 0.01	0.936
C:10 , γ: 1	97.5		
C:10 , γ: 0.01	94		

- Noisy data
  Feature vectors which are selected in this experiment are suction pressure (P_{suc}), compressor speed (V_{cpr}), superheat temperature (T_{sh}), and proportional compressor mass flow rate ($K_{\dot{m}}$). For emulating noise to the data, random noise with normal distribution $\mathbb{N}(0, 2)$ is added to T_{sh} and deducted from evaporation temperature (T_0). This range is selected so that the noise affects T_{sh} in range $[-4, 4]°C$. T_{sh} is calculated as

$$T_{sh} = T_{suc} - T_0 \tag{9}$$

  Afterwards, P_{suc} is calculated by the thermodynamic behaviour of noisy T_0 and $K_{\dot{m}}$ is affected by noisy T_{suc}. Assume that compressor mass flow rate ($\dot{m}$) is calculated as

$$\dot{m} = \rho V_{cpr} A \tag{10}$$

  where ρ and A are refrigerant density at the inlet of the compressor, and the cross-sectional area of the compressor inlet, respectively. However, in this experiment, $K_{\dot{m}}$ is used instead of $\dot{m}$ in order to ensure that the data is independent of the components' type and physical characteristics. Therefore, $K_{\dot{m}}$ is calculated as

$$K_{\dot{m}} = \rho V_{cpr} \tag{11}$$

  In this regards noisy T_{suc} and P_{suc} affect ρ as

$$\rho = \frac{M_m P_{suc}}{R T_0} \tag{12}$$

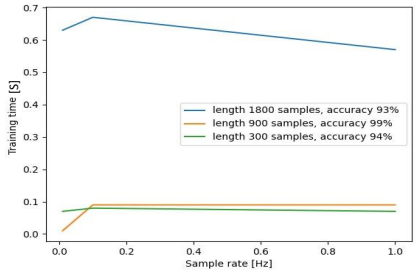

Fig. 33: Data with different sample rates and different lengths are fed into the SVM classifier. The results of the classification appear in the legend showing that input data with 900 samples achieved the most classification accuracy while training time is constant when changing the sample rate from 0.01 to 1 Hz. In addition, when selecting data with 900 samples, training time is six times less than selecting data with 1800 samples.

where M_m and R are molar mass and ideal constant of gas refrigerant, respectively. Fig. 34 represents an example of added noise to the data. The SVM classification at right shows that noisy data is classified with high accuracy.

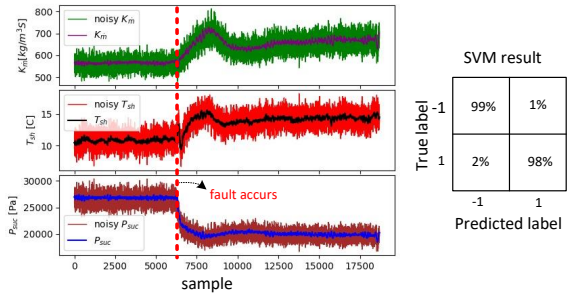

Fig. 34: In Paper B, noise is added to T_{sh} and reduced from T_0. Noisy data is visualised. Noise is not added to V_{cpr} since the dynamic of the compressor speed is lower than getting effect from the noise. The confusion matrix at right shows 99% and 98% accuracy for detecting non-faulty conditions (label -1) and defective evaporator fan conditions (label 1), respectively.

- Perturbed data
  Refrigeration systems run in different operating conditions and/or with different system configurations. Therefore, data may vary from system to system. In this experiment, temperature perturbation is applied to the measurements to emulate various data, covering more system variation and/or operational conditions. In

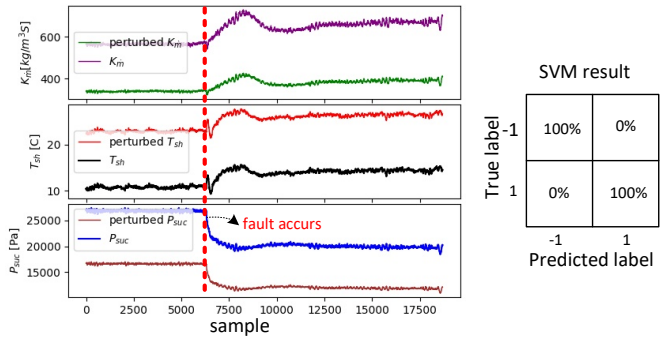

Fig. 35: In this example, T_{sh} is increased by 12 °C and T_0 is decreased by the same value. This offset value is not added to V_{cpr} as different refrigeration systems can operate with different compressor speeds. The confusion matrix at right indicates 100% classification accuracy when the correlation of the data points is changed by some perturbation. This experiment is used in Paper B.

paper B, T_{sh} is negatively offset to 5 degree C and positively offset up to 12 °C. Those values approximately represent minimum and maximum acceptable T_{sh} for different operational conditions. $K_{\dot{m}}$ and P_{suc}, and V_{cpr} are affected accordingly. Fig. 35 represents an example in which T_{sh} is increased by 12 °C. The SVM can classify perturbed data with 100% accuracy even though V_{cpr} remains unchanged, and the correlation between data is different from training data.

- Disturbance on operation
  When the temperature of the cold room is the same as the setpoint, the compressor stops working until the changes in the cold room temperature exceed the valid value. Then the compressor starts working again to keep the cold room temperature at the setpoint temperature constantly. The original data which is collected in the laboratory contains only the running compressor condition. Therefore, a periodic disturbance is added to the data so that we emulate both stops and running operational modes of the compressor, see fig. 36. A disturbance Δ with a period of 300 s is added to P_{suc}. T_{sh_Θ}, P_{suc_Θ},V_{cpr_Θ}, and $K_{\dot{m}_\Theta}$, which are feature vectors influenced by the disturbance as follows:

$$
\begin{aligned}
\underline{\Delta} &= 0.5(\sin\left(\frac{2\pi t}{300}\right) + 1) \\
P_{suc_\Theta} &= P_{suc} + \underline{\Delta} P_{suc} \\
T_{suc_\Theta} &= T_{suc} + \Delta T_{suc} \\
T_{sh_\Theta} &= T_{suc_\Theta} - T_0
\end{aligned}
\tag{13}
$$

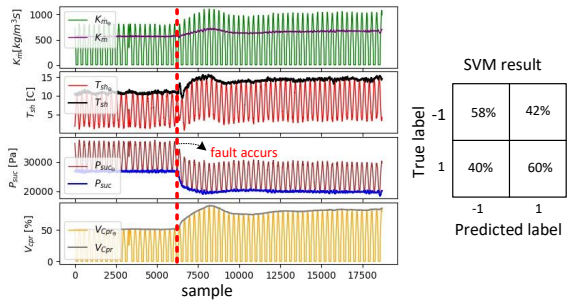

Fig. 36: In this example, a disturbance is applied to all four feature vectors as presented in Experiment 7.4. SVM result is represented at right where non-faulty condition (label -1) and faulty condition (label 1) are detected with 58% and 60% accuracy, respectively. This experiment is used in Paper B.

$$V_{cpr_\Theta} = \begin{cases} \underline{\Delta} V_{cpr}\ V_{cpr} & \Delta > 0.3 \\ 0 & otherwise \end{cases} \tag{14}$$

$$K_{\dot{m}_\Theta} = \rho_\Theta\ V_{cpr_\Theta} \tag{15}$$

In Fig. 36, the SVM classification result at the right is not satisfactory, which means that the On/Off mode of the compressor has made some data overlap when the compressor is in off mode, which decreased the classification accuracy. However, Table B.3 indicates that PCA-SVM can improve the classification accuracy even though the data consists of both stopping and running compressor mode.

Experiment 6

Paper C provides data from 21 classes, including 20 classes of different faults in refrigeration systems and one non-faulty condition. SVM classifier is used for fault detection and diagnosis. In Table 9, OVO and OVR methodologies are compared while the hyperparameter C is selected as $C = 10$ in both tests. Fig. 37 represents the SVM classification responses for 11-dimensional verification data using both OVO and OVR functions. In this figure, *True labels* are the allocated labels to the classes, and *Predicted labels* are the estimated output by the classifier. Thus, the values in the diagonal are correct estimations by the classifier in the range [0,1]. Accordingly, every non-zero value out of the diagonal is the wrong estimation of the classifier. According to Fig. 37 and Table 9, OVO achieved more satisfactory results than OVR even though the estimation time in OVO is higher than OVR.

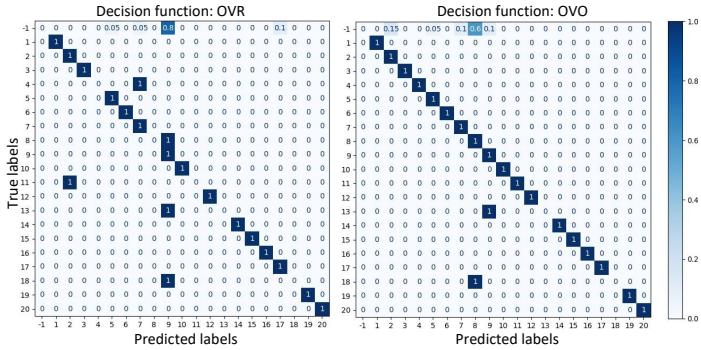

Fig. 37: The SVM response where hyperparameters are: C= 10, $\gamma = 0.01$, $kernel = RBF$, and OVO decision function, in the right, and SVM response where hyperparameters are: C= 10, and OVR decision function, in the left. OVO represents a more accurate classification for most of the classes using unseen data.

Table 9: Comparison of OVO and OVR for the SVM classifiers in Fig. 37 shows that OVO are the better approach in our case regarding classification accuracy though the estimation time is more than 3 times of the estimation time using OVR approach.

method	accuracy	training time	estimation time
OVO	85%	6.1 s	0.7 s
OVR	71%	7.1 s	0.2 s

Experiment 7

14-dimensional data is recorded, which are named in Subsection 7.2. Each of the 14 feature vectors is tested to see if it helps higher classification accuracy or adversely affects the classification estimation. To look at the experiment in detail, SVM classification using the original 14-dimensional data is represented in Fig. 39 that achieves 71% accuracy. Among all the feature vectors, three feature vectors, T_{amb}, P_{cpr}, ρ, are found more interesting as they either have no effect or an adverse effect on classification result. Fig. 38 illustrates the effect of each of those three feature vectors on the classification results. Dropping P_{cpr}, has almost no effect on classification result as we achieved almost the same classification accuracy as using 14-dimensional data. Dropping ρ slightly improves the classification accuracy to 76%. However, dropping T_{amb} improves the classification result comparably to 86%. Fig. 39 illustrates the classification result using 11-dimensional data where all three feature vectors mentioned above are removed. This classifier achieved 87% accuracy, which is improved a lot compared to the 14-dimensional classifier. The false positive rate is also improved compared to the other tests. Note

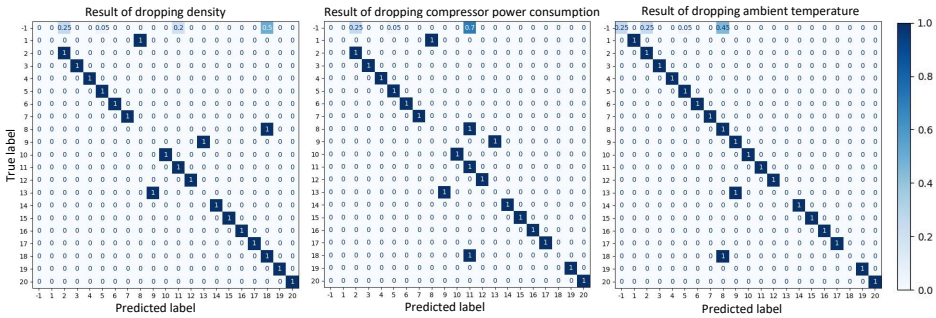

Fig. 38: The three confusion matrices illustrate the classification of 21 classes using different input data. The input data is 13-dimensional in each test, where the dropped feature vector is noted at the top of the confusion matrices.

that SVM classifiers in this experiment have the same hyperparameters: $C = 1000$, $\gamma = 0.01$, $kernel = RBF$, and OVO decision function.

Experiment 8

This Experiment illustrates the response of the training phase for SVM, LDA, and LDA-SVM algorithms in Table 10. The input data used for this experiment is 11-dimensional data specified in Experiment 7 and each algorithm is specified in Table 17 in Subsection 8.2. These trained models are used in Paper C, Appendix 1.4 to classify unseen verification data in a different operational condition.

Table 10: Responses of the classifiers for training and test phase showing the better results obtained by SVM and LDA-SVM. High false positive rates mean that the classifiers are not well trained to identifying non-faulty condition among the other faulty conditions.

algorithm	accuracy	false positive rate
SVM	91%	60%
LDA	70%	69%
LDA-SVM	91%	87%

Experiment 9

An LDA algorithm is used in Paper C for dimensionality reduction and classification. In order to reduce the dimensions, eigenvalues of matrix Ω are required to evaluate how many dimensions are necessary. Fig. 40 presents the ratio of each eigenvalue against the sum of eigenvalues calculated from matrix Ω in descending order. This figure indicates

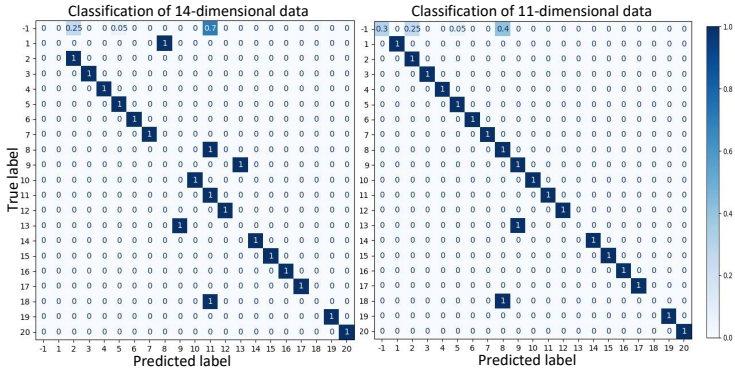

Fig. 39: The confusion matrix at left proposes 11-dimensional data to SVM, in which T_{amb}, P_{cpr}, ρ are removed. The confusion matrix at right shows the classification response using original data with 14 feature vectors. Diagonal values represent the correct classification responses.

Table 11: Comparison of LDA classifiers when transforming data into different dimensional space. This comparison shows a better classification accuracy when using 6 and 8 LDs for classification. However, looking to the training time and estimation time using 6 LDs is more suitable than 8 LDs.

dimension	accuracy	training time	estimation time
4	61%	$0.01s$	$0.3\ s$
6	81%	$0.02s$	$0.3\ s$
8	81%	$0.04s$	$0.4\ s$

that the first 6 eigenvectors in descending order represent the most data characteristics. Table 11 represents the results of the LDA when data is transformed from 11 dimensions into different dimensions and then classified by the LDA. The results indicate that the 6-dimensional data achieves the best result with 81% classification accuracy and LDA, in general, performs poorly regarding detecting non-faulty class. Higher dimensional LDA represents the same classification accuracy as 6-dimensional LDA, and lower dimensional classifiers perform poorly. Table 12 represents the results of different LDA-SVM classifiers, which are the same in hyperparameters and different in input dimensions. 4-dimensional LDA-SVM has the same total accuracy as 6-dimensional LDA-SVM. However, they have a big difference in false positive rates. Therefore, we consider 6-dimensional LDA-SVM more accurate than 4-dimensional as the classification of non-faulty class matters a lot among all classes. 8-dimensional LDA-SVM has higher total accuracy than 6-dimensional; However, the false positive rate is higher compared to the 6-dimensional LDA-SVM classifier. Therefore, we suggest 6-dimensional LDA-SVM as the one that performs the best among the other LDA-SVM models.

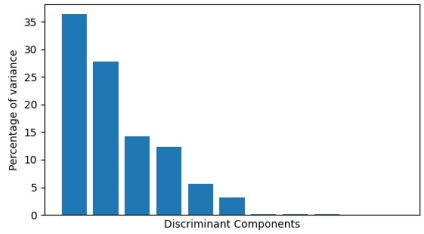

Fig. 40: Variation of discriminant components (LDs) in descending order indicates that the first six eigenvectors carry the most information of data.

Table 12: Comparison of LDA-SVM classifiers when transforming data from 11 into 4, 6, and 8 dimensions before classifying data by SVM. This comparison reveals that reducing input dimensions to 6 before classifying input data by SVM is the best choice concerning false positive rate. However, regardless of the false positive rate consideration, LDA-SVM with 8 LDs are the best choice.

LDA dimension	accuracy	training time	estimation time	false positive rate
4	86%	$0.7s$	1.5 s	100%
6	86%	$0.7s$	1.5 s	0%
8	90%	$0.8s$	1.5 s	13%

8 Results

This chapter consists of two binary and multi-class classification parts. Each part represents the results of deep learning classifiers and shallow learning classifiers. This chapter summarises the results of the papers done during the PhD program. An important correction in Paper C is investigated and notified to the publisher. The corrected results are presented here and the corrected sections of Paper C can be found in Appendix 1.

8.1 Binary classification

Papers A and B investigate fault detection in supermarket refrigeration systems. In addition, these papers used the same data to evaluate shallow learning and deep learning algorithms.

CNN model sensitivity evaluation

CNN is selected and evaluated as a deep learning model in Paper A. The model used in this paper is presented in Table 5. Paper A argues the model sensitivity against various sample rates, number of samples, noise and perturbation in validation data.

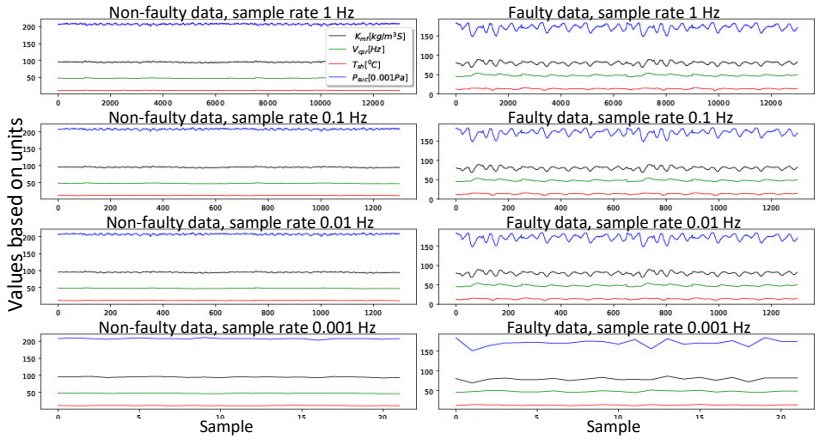

Fig. 41: Visualisation of data down-sampling in non-faulty and faulty conditions from 1 Hz to 0.001 Hz in Paper A. These plots show that data information and characteristics are preserved by down-sampling of the data to 0.01 Hz.

- Sensitivity against sample rate and data log size
  The effect of sample rate and the number of samples during the training process on the classifier's performance are studied. The original data taken in the laboratory is recorded with a 1 Hz sample rate. Then, data resolution is reduced for this experiment. Fig. 41 visualizes an example of non-faulty and faulty data where down-sampling is done from 1 Hz to 0.001 Hz. In this experiment, decreasing sample rates from 1 Hz to 0.01 Hz does not affect the data characteristics. However, down-sampling data to 0.001 Hz cannot preserve the important information of the data as the data characteristics are changed. These data with different sample rates are fed into the CNN algorithm. The classification accuracy and loss for seven experiments are introduced in Fig. 42, where the input size and resolutions are different for each experiment.

  Due to the limited number of samples in each data log, the number of samples decreased when we lowered the sample rate. Therefore, we investigate the effect of the number of samples and sample rate on the training process. Fig. 42 shows that the original dataset, which is indicated by blue colour, obtains the best response, where the model trained to classify the data with high accuracy after about 25 epochs, and the loss is low compared to the other experiments. Comparing orange and red experiments (the experiments that input data have the same sample rate but different numbers of samples), the orange experiment with a higher number of samples performs better. Even though the orange and red experiments finally present almost the same accuracy, the algorithm requires more iteration to be trained well, and the loss value is higher in red than in the orange experiment. This can be due to the lack of information when the number of samples is insufficient.

  Another observation in Fig. 42 is the comparison of the experiments with the same number of samples but different sample rates. Comparing green and orange experiments (the experiments that input data have the same number of samples but different sample rates), both tests result in the same accuracy and loss. However, the green experiment, in which the data has a lower sample rate, could be trained faster than the orange one. Fig. 43 explains the reason why the CNN classifier is trained faster when data has a lower sample rate.

  Fig. 43 shows one dataset with two different sample rates. The figure at the bottom has a lower sample rate, and more oscillations can be observed. This explains the low thermodynamic behaviour of refrigeration systems. In this figure, 200 samples taken every 10 s or 0.1 Hz contain more information than 200 samples taken every second. Therefore, the training process converges faster when each dataset contains more information. Other experiments behave the same, e.g. purple and red. To conclude the observations in these experiments, the number of samples has more effects than the sample rate to have a faster convergence during the training process, and a lower sample rate is more effective than a higher sample

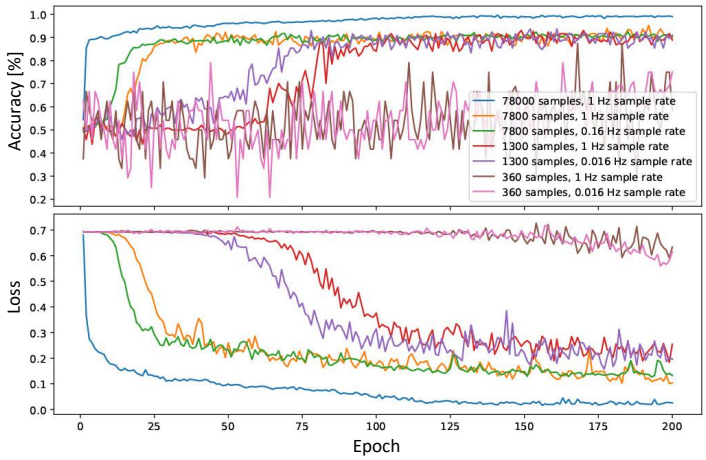

Fig. 42: The results of CNN classifier in Paper A, using validation data in different resolutions and lengths, which are specified in the legend. The model with a lower number of samples requires more iterations than the one with more samples for training. However, when two datasets have the same length, the one with a lower sample rate converges faster.

rate until the information of the thermodynamic behaviour is not disappeared.

- Sensitivity against noise To evaluate the model sensitivity, the CNN model is run 100 times with different random noises within the range $\mathcal{N}(0, 2)$. Noise is added to the validation data, as explained in Experiment 5 in Subsection 7.4. The result is presented in Fig. 44. This figure indicates higher than 99% classification accuracy for non-faulty data for all 100 runs. Thus, the model is reliable with less than a 1% false positive. In addition, for 95 runs out of 100, faulty data is classified with more than 97% accuracy.

- Sensitivity against data perturbation Perturbation of the validation data is done to evaluate the model's sensitivity when system operating conditions differ. Data perturbation is explained in Experiment 5, in Subsection 7.4. The result of 100 runs using different random perturbations in the range [-3,3.5] is shown in Fig. 45. It is investigated that non-faulty data is classified with 99% accuracy for 92 out of 100 runs. For the rest of the runs, the classification accuracy is less than 91%. Therefore, a 1% false positive for the model is reliable 92% of the time when validation data correlation is different from training data. The model shows a satisfactory result for detecting faulty data. In all 100 runs, the model classifies the faulty data with 99% accuracy.

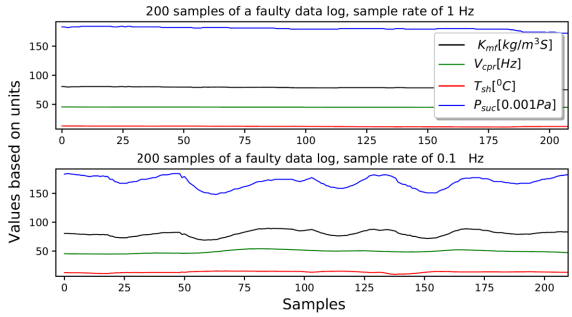

Fig. 43: For the sample rate test in Paper A, a dataset is zoomed to 200 samples. The original data log at the top has a 1 Hz sample rate. In the same data log, the sample rate is changed to 0.1 Hz, and then 200 samples are visualized at the bottom.

SVM model specification

Paper B studies the SVM classifier for binary classification. After several parameter optimisations, the best model is specified with RBF kernel function where parameters C and γ are determined as 100 and 1, respectively. Input dimensionality reduction is investigated manually and automatically because less computation and less required information are preferred. In Paper B, three models are made which are different in input dimensions. First, 14-dimensional data are fed into the model. Second, four feature vectors most relevant to the fan fault detection problem are selected based on the expert's knowledge. Third, PCA is used for dimensionality reduction from 14 to 2 dimensions. All these models with different input dimensions achieved the same result of 98% accuracy in training and 100% accuracy in validation.

SVM model sensitivity evaluation

Even though all 3 SVM models mentioned above achieved the same results with the same training and validation data set, we prefer to evaluate less dimensional input classifiers as it might be computationally more efficient and eventually cheaper for implementation. Therefore, the sensitivity of a 4-dimensional SVM and a PCA-SVM are investigated below.

- Training data sensitivity against down-sampling and data length Table. 13 argues the effects of data resolution and the data size during the training process. In this test, 4-dimensional data are fed into the SVM model. As it is observed, down-sampling does not affect the SVM classification until the dynamic characteristics of the data are preserved. However, data length has a high impact on training.

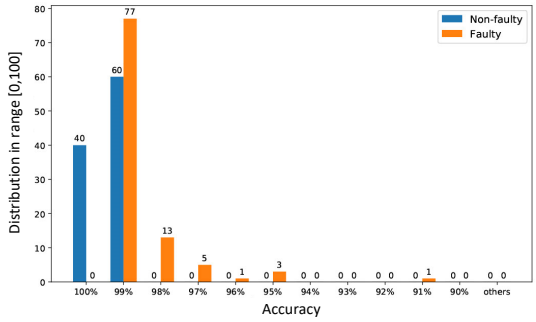

Fig. 44: Distribution of accuracy for 100 runs of the CNN classifier in Paper A, where different random noise is added for each run to the validation data. The accuracy distribution for the non-faulty data classification is determined by blue, and the accuracy distribution of the evaporator fan fault detection is presented by orange.

This test illustrates the proper size of the data for SVM classification, which is 900 samples for this model. An insufficient number of samples leads to under-fitting the model. Too many sample points have a negative impact both on computation time and accuracy. Looking through Table 13, when the number of samples doubles from 900 to 1800, the computation time increases at least 60% for the input data with the same sample rate [48].

- Sensitivity against validation data resolution In this test, all 3 SVM classifiers are trained with high-resolution data with a sample rate of 1 Hz. Then, the validation data is down-sampled to 0.1 and 0.01 Hz. However, the same results are obtained. This result illustrates that SVM classifies the data independent from the data sample rate until the required dynamic behaviour in the data is preserved. This test shows that the 0.01 Hz sample rate is still good enough for classification using SVM.

- SVM sensitivity to the validation data variation In this observation, the sensitivity of the SVM model is evaluated when validation data is noisy, perturbed, or data is taken from the system in On/Off mode. In refrigeration systems, the system operates in on/off mode when low cooling capacity is needed in the cold room; In other words, when the cold room temperature reaches the set point. Therefore, the compressor alternate between on and off mode to keep the cold room temperature at the set point. Table 14 represents the result of several tests on all 3 SVM classifiers, which are different in the input dimensions. In addition, PCA-SVM results for validation data variation are introduced in Fig. 46. As it appears in this figure, PCA can deal with noise and perturbation well. Thus, PCA-SVM

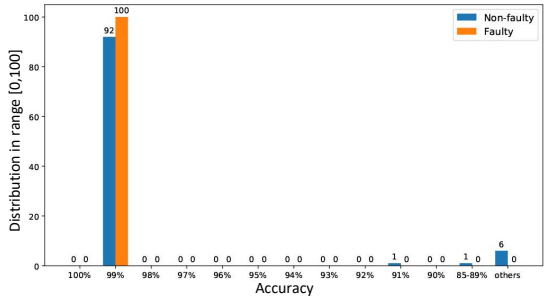

Fig. 45: Distribution of accuracy for 100 runs of the CNN classifier in Paper A, where the validation data is randomly perturbed for each run. Here, the orange block shows that the evaporator fan fault is diagnosed with 99% accuracy for all 100 runs and the blue blocks show the accuracy distribution of non-faulty class classification.

can classify validation data with noise or perturbation with high accuracy, the same or even better than 4-dimensional or 14-dimensional SVM. The shape of the data in On/Off mode is changed after using PCA because the data correlation is different when the compressor stops operating. Therefore, PCA-SVM can not correctly classify the data in On/Off mode as in operating mode. However, PCA-SVM classifies data much better than 14-dimensional and 4-dimensional SVM, see Table 14.

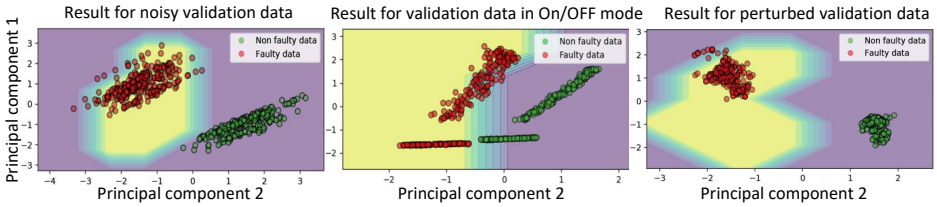

Fig. 46: Visualisation of PCA-SVM classifications using different validation data according to Paper B. The colours, from yellow to purple, show the contoured space that separates each class. So, the classifier takes an optimal position between these two colours in the contoured space such that it classifies the data with the lowest loss.

Table 13: The result of SVM model classification in paper B, where down-sampling and data shortening are applied. Here, SVM obtained a better classification result using data with 900 samples. Training time of using 900 samples; however, reducing the number of samples from 900 to 300 doesn't affect the training time compared to decreasing the number of samples from 1800 to 900.

length	sample rate $[Hz]$	training time (s)	accuracy [%]
300	1	0.07	94
	0.1	0.08	94
	0.01	0.07	94
900	1	0.09	99
	0.1	0.09	99
	0.01	0.1	99
1800	1	0.57	93
	0.1	0.65	93
	0.01	0.63	93

8.2 Multi-class classification

In Paper C, different faults are applied to the high-fidelity simulation model for collecting data in different faulty conditions. The model and types of faults are described in Subsection 4.3. Paper C proposes different algorithms for classifying 21 classes, including a non-faulty class and 20 classes of different faults. LDA and PCA are proposed for dimensionality reduction. CNN, SVM and LDA models are used for classification purposes. In Paper C, 12-dimensional and 14-dimensional data are proposed, described in 7.2.

ML algorithms mentioned in Paper C are tested first Using the 12-dimensional input data. The training responses can be found in Table 15, showing high classification accuracy during the training phase. In this table, LDA and SVM obtained the best results with better than 99.6% classification accuracy and 0% false positive. On the other side, PCA-SVM gives an unacceptable result as the classification accuracy is 55.4%. The CNN performs poorly in detecting the non-faulty class with a 32% false positive rate in the training phase. Table 15 compares computation time for LDA, SVM, LDA-SVM and CNN. Even though LDA processes 5-dimensional input data, SVM is trained faster than LDA. The verification of these classifiers is studied using data in another operating condition. Therefore, in Paper C, verification data is collected for all classes where T_{set} is set to 4 °C, heat load to 13 kW, and ambient temperature to 17 °C. The range of operating conditions for training data is described in Section 7.2. However, from Table 16, the models can not classify data from unseen operation conditions properly as the classification accuracies and false positive rats dropped remarkably compared to the training responses. Thus, we need more investigation to optimise the performance of the classifiers.

Table 14: The sensitivity comparison of 3 different SVM classifiers proposed in Paper B to the validation data variations. Here, PCA-SVM outperforms the other models, in all three noise, on/off and perturbation tests, for classification of both non-faulty and faulty data.

	algorithm	non-faulty[%]	faulty[%]	operation time[s]
noisy	14D SVM	98.5 -99.6	98 -99.4	0.31
	4D SVM	98 -100	98 -99.4	0.24
	PCA-SVM	98 -100	98 -99.6	0.25
on/off	14D SVM	50-60	53-60.5	0.33
	4D SVM	55-60	54-61	0.25
	PCA-SVM	85-86	95.5-96.4	0.25
perturbed	14D SVM	89-100	97-100	0.32
	4D SVM	99.2-100	99-100	0.24
	PCA-SVM	100	100	0.23

Table 15: Comparison of the classifiers in training phase in Paper C using 12-dimensional data. This Table indicates the better performance of LDA and SVM regarding non-faulty data classification during the training phase. CNN performs poorly to classify non-faulty data and PCA-SVM result of classification is not satisfactory. LDA-SVM obtains high classification accuracy for faulty classes but performs poorly for non-faulty class classification.

algorithm	dimensions	accuracy	false positive	training time
SVM	12	99.6%	0%	1.1 s
LDA	12 to 5	99.8%	0%	3.2 s
CNN	12	94%	68%	112.5 s
PCA-SVM	12 to 2	55.4%	24%	7.2 s
LDA-SVM	12 to 5	96.6%	18%	1 s

Sensitivity of the classifier against different operating conditions

As the classifiers mentioned in Table 16 could not deal with the verification test satisfactorily, we decided to improve the models by increasing the input information. Therefore, another dataset is collected with more excitation during system operation and two new feature vectors namely T_{set} and T_{amb} are added to the input data. Ambient temperature in range [10,30]°C is added to the limitations specified in Subsection 7.2. Therefore, the 14-dimensional data holds some variation of ambient temperature as the excitation of operation conditions. The results of using more excited data sets for training of the algorithms are studied in, Experiment 8 in Subsection 7.4. To optimize the performance of the classifiers, Experiment 7 in Subsection 7.4, is done to investigate if one or more feature vectors may have a negative impact on classification accuracy. Experiment 7

Table 16: Comparison of the classifiers in verification test where the training is evaluated in Table 15. These results are not satisfactory and motivate us to enrich training data to have better trained models.

algorithm	verification accuracy	false positive
SVM	76%	92%
LDA	52%	100%
CNN	53%	91%
PCA-SVM	48%	96%
LDA-SVM	57%	100%

illustrates an investigation of three feature vectors: compressor power consumption, density, and ambient temperature which show adverse impacts on the classification results. Finally, the algorithms described in Table 17 can classify 21 classes with the highest accuracy. After training the algorithms described in Table 10, The model verified with an

Table 17: Classifiers specifications with different hyperparameters and input data in Paper C. Here, df stands for decision function, kf for kernel function, and LD for Linear discriminants.

algorithm	specification	dropped features
SVM	df=OVO, kf=RBF , C=1000 and $\gamma = 0.01$	T_{amb}, P_{cpr}, ρ
LDA-SVM	LD=6 df=OVO, kf=RBF, C=1000 and $\gamma = 0.01$	T_{amb}, P_{cpr}, ρ
LDA	LD = 6	T_{amb},P_{cpr}, ρ

unseen data set with the same dimensions as training data. The results of the classifications are illustrated in Table. 18. Fig. 47 represents the models' responses in detail. The LDA-SVM model stands alone for the classification of the non-faulty condition with a 0% false positive rate and very close classification accuracy compared to SVM. However, the computation time for decision-making is three times more than SVM. LDA-SVM classifies 17 faults out of 20 with 100% accuracy. Three misclassified faults are namely blocked expansion valve, loose expansion valve, and return temperature with positive offset. The SVM model performs fault localization better than the other models, which can classify 18 faults out of 20 with 100% accuracy. The only misclassified faults are the blocked expansion valve fault and return temperature sensor fault with negative offset, which are misclassified with loose expansion valve fault and evaporator fan low performance, respectively. However, SVM is not able to detect the non-faulty condition as it obtained a 90% false positive rate. Among the mentioned models, LDA performs with lower accuracy of 80% and detects 16 faults out of 20 simultaneously. The computation time in LDA is less than the other two methods. However, the performance is not as satisfactory, neither for non-faulty condition detection nor for fault localization.

Table 18: Comparison of the models in Paper C (correction part). This table indicates a better performance of SVM regarding accuracy and the best performance for LDA-SVM considering false positive rate.

Model	accuracy	false positive rate	training time	prediction time
SVM	87%	70%	2.6 s	0.4 s
LDA-SVM	86%	0%	0.7 s	1.5 s
LDA	81%	100%	0.02 s	0.3 S

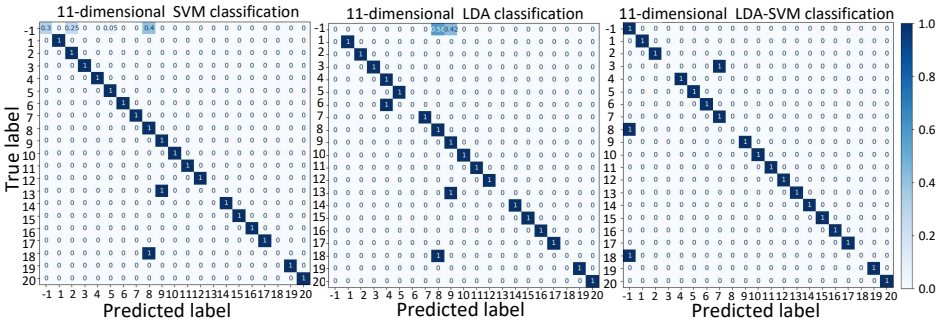

Fig. 47: The verification results after training algorithms by more excited data. The correct classifications are shown by diagonal values. Label -1 indicates non-faulty class and label 1 to 20 represent faulty classes which are described in Table 3. This results are included in the correction part of the Paper C.

9 Contributions

This chapter represents the main contributions of Papers A, B, and C.

Developments of fault detection tools are aimed at Bitzer electronics as these tools can potentially affect the products' efficiency, reliability, and uptime. This study aims to investigate the capability of some of the machine learning methodologies in the field of refrigeration systems, which can be improved by increasing available data at Bitzer electronics through its digital network. This chapter summarizes the contributions of Papers A, B, and C to the problem of automatic fault detection and diagnosis in refrigeration systems. This study tries to look at the problem from an industrial point of view. Therefore, the evaluation of the proposed models has been done so that the refrigeration industry can benefit from this investigation. In this regard, computation time, data resolution, false positive rate, number of required samples, and ability to lower the cost for human resources are considered.

9.1 Evaporator fan fault detection

Paper A uses a deep learning model to argue for evaporator fan fault detection in refrigeration systems connected to the supermarket condensing unit. Bitzer produces condensing units for supermarket refrigeration systems called ECOSTAR. Ecostar controls the compression and condensation phases of the refrigeration process. In the case of using data from the Ecostar controller, information from the evaporation and expansion side of supermarket refrigeration systems is not available. However, evaporator fan fault detection may affect Bitzer products' uptime and reliability due to increasing compressor efficiency or severity. Thus, detecting such a fault, which can indirectly affect the products' reliability, is an achievement. Therefore, CNN is proposed to use the data from an ECOSTAR unit to detect the defective evaporator fan.

In this way, real data from a refrigeration system is collected at the Bitzer electronics laboratory for training and verification of the CNN model. The data structure in the field may vary regarding operational conditions, variation of components, data sample rates, data length, noise, disturbance etc. Thus, the model's sensitivity against some of these variations is investigated, such as sensitivity against noisy data, variation of sample rate, data length, data perturbation and disturbance. Remark that in Paper A, perturbation is introduced to investigate correlations between the data features since different components and loads in supermarket refrigeration systems can affect data structure.

The results of 100 different verification tests on perturbed and noisy validation data indicate that the model can distinguish the evaporator fan fault with better than 98% classification accuracy for 92 runs out of 100 runs. The field applications vary in length of data logs and sample rates due to the different requirements for embedded hardware and software. In paper A, data with various sample rates and lengths are tested, showing

limitations on data length and lowering sample rates.

Besides the CNN, which is a deep-learning neural network model, Paper B embarks on several shallow-learning neural network models for evaporator fault detection. When designing a fault detection algorithm in the field, computation time and space in embedded software and hardware are of vital importance. Thus, it is worth investigating if there is a shallow neural network which can contribute to the problem of fault detection in refrigeration systems with a satisfactory result. Paper B introduces SVM as a classifier and compares three models with different input dimensions regarding their accuracy and computation time. Paper B uses input data in three different shapes: high-dimensional input data, which includes signals from the controller; 4-dimensional data, in which refrigeration systems experts select the features; and 2-dimensional data, in which data dimension is reduced using PCA.

The results show that 2-dimensional PCA-SVM, where the dimensions are reduced to 2 by PCA, is computationally more efficient than 4-dimensional SVM. In addition, the 4-dimensional SVM in Paper B requires careful input data selection to have the most relevant information for a specific fault classification, which is inefficient and expensive regarding experts' effort compared to the PCA-SVM. Moreover, the 2-dimensional PCA-SVM model can obtain nearly the same results as 4-dimensional SVM and full-dimensional SVM when feeding noisy and perturbed data for verification tests. However, it obtained a higher classification accuracy when the system ran in ON/OFF mode. Thus, PCA can be used as a form of data normalization and optimizes computation time when using the SVM classifier. Paper B indicates that a PCA-SVM trained in a specific sample rate can be used to classify the fault in various refrigeration systems with different sample rates. However, the classifier is sensitive to the data length. Too many samples are not efficient, as the classifier would struggle with more noise and outliers during training.

9.2 Multi-fault detection and diagnosis

Regardless of the evaporator fan fault detection done in Papers A and B, comparing and evaluating the classifiers for other possible faults in refrigeration systems is essential. Paper C presents twenty types of possible faults in refrigeration systems: sensor faults and main components faults. Then, different classifiers are used to classify twenty-one data classes, including non-faulty data. In this paper, it is decided to simulate the faults in a high-fidelity Simscape model of refrigeration systems because emulating several faults in the laboratory setup is expensive and time-consuming. Second, we are unsure if the labelled data from the field, already in Bitzer electronics, are correctly labelled according to our purpose. Finally, simulation data can be a proper replacement for training such systems when systems' operational conditions and types of components are varied. Thus, generating data from the simulation model is trustful, feasible and cheaper to cover most operational conditions and systems variations for training purposes. In

Paper C, several algorithms are proposed: CNN, SVM, LDA, LDA-SVM and PCA-SVM. Then, the classifiers are evaluated based on classification accuracy, false positive rate, and computation time. Each model is used to classify all types of faults simultaneously. Paper C compares a deep neural network model and some shallow neural network models such as SVM and LDA.

Paper C emphasizes the careful selection of data variation, as low data variation may lead to a completely different result for the same model. In addition, the selection of data features is important as some of the data features can negatively affect the classification results. In Paper C, dropping ambient temperature, density calculated by suction pressure and suction temperature at the inlet of the compressor, and power consumption of the compressor improved the classification results remarkably.

Even though the CNN model and PCA-SVM model could achieve satisfactory results for binary classification in Papers A and B, the PCA-SVM model diagnoses multi-faults poorly, and the CNN model is unreliable with a high false positive rate. Thus, PCA can not perform as well as LDA for dimensionality reduction when the classifier should deal with several classes. Among the LDA-SVM, SVM, and LDA classifiers, the SVM and LDA-SVM models obtain the best performance addressing multi-fault classification in refrigeration systems. The LDA-SVM performs well for detecting non-faulty class from faulty with 0% false positive rate, and SVM localizes 18 faults out of 20 with 100% accuracy. The computation time for predicting the label is less than one-third for the SVM algorithm compared to the LDA-SVM. Remark that the SVM model performs weaker than the LDA-SVM, counting for the false positive rate.

10 Closing remarks and perspectives

In this chapter, we conclude that how data-driven models perform when addressing fault detection and diagnosis in refrigeration systems from an industrial perspective. Furthermore, some of the remaining questions about the problem in the field of refrigeration systems are introduced.

10.1 Industrial remarks

Faults in industrial refrigeration systems are inspected by technicians when the symptoms of the faults, e.g. inconsistent cold room temperature, extra noise from the system, abnormal behaviour of some components such as continuous work of compressor or fan appear. However, in some faulty cases, refrigeration systems operate inefficiently for a long time, leading to some fatigue faults. In many industrial applications, a set of alarms exists which notify users about the symptoms if the system operates out of the system limits. However, the faults can not diagnose by the alarms and sometimes, the faulty systems run for a long time without reaching the system limits. Therefore, early and automatic FDD in refrigeration systems is attractive for manufacturers and customers.

Using all signals as the inputs of the classifier may be impractical in real-life applications, as all signals might not be available in various systems. In the case of binary classification, PCA ensures that the essential information is still captured from the data for the fault classification and that it handles perturbations better than using original data. Normalizing data using the PCA algorithm improves the fault detection robustness against refrigeration systems variations. In addition, the PCA-SVM model can be separated into two phases: dimensionality reduction and classification. Then, the first phase is feasible to be applied to the embedded controller; then, two-dimensional data can be sent to the third party for fault classification.

In this study, offline classification is also offered. A trained model can be applied to the controller, which requires less processor capacity and is safer because the model cannot be trained with wrongly labelled data.

In industrial applications, a low false positive rate for FDD methods is very important. A misclassification of a functional system as a malfunction is expensive and decreases the reliability of the FDD tool. Paper C states the LDA-SVM model with 0% false positive capability. Thus, LDA-SVM is emphasized as the reliable model for detecting the fault, while the SVM model can be used to diagnose the specific fault in the system with high accuracy.

10.2 Perspectives

In this research, the faults are defined with narrow limitations. However, the research can be developed by covering a broader definition of the faults. In addition, in Paper

C, the transient part of the data is not considered. Even though in many industrial applications such as reefer containers and cold warehouses, the refrigeration systems often run in steady-state, The author appreciates more investigations in which a transient part of the data is analyzed.

Of importance is to verify the models with real data from the field. This research can give the industry an overview of the applicability of machine learning-based models for FDD purposes in refrigeration systems. However, more investigation is required to find the best model or model parameters before applying the model to refrigeration systems, which requires reliable data from the field. The FDD model can be improved and developed in the future by applying more data from the field.

Last but not least, before applying any FDD models mentioned above to the system, the processing requirements need to be calculated. The result of the required processor space significantly impacts selecting the best model due to the industry's limited processor space. Therefore, according to Paper C, the selected model must have a trade-off between a high accuracy, low computation, and low false positive rate.

References

[1] R. Badia-Melis, U. Mc Carthy, L. Ruiz-Garcia, J. Garcia-Hierro, and J. Robla Villalba, "New trends in cold chain monitoring applications - a review," *Food Control*, vol. 86, pp. 170–182, 2018.

[2] S. Martinez, J. M. Murguia, B. Rejas, and S. Winters, "Refrigeration and child growth: What is the connection?" *Maternal and child nutrition*, vol. 17, no. 2, pp. e13 083–n/a, 2021.

[3] D. Coulomb, "Refrigeration and cold chain serving the global food industry and creating a better future: two key iir challenges for improved health and environment," *Trends in food science & technology*, vol. 19, no. 8, pp. 413–417, 2008.

[4] J. H. Hanks and R. E. Wallace, "Relation of oxygen and temperature in the preservation of tissues by refrigeration," *Proceedings of the Society for Experimental Biology and Medicine*, vol. 71, no. 2, pp. 196–200, 1949.

[5] A. F. Santos, P. D. Gaspar, and H. J. L. de Souza, "Refrigeration of covid-19 vaccines: Ideal storage characteristics, energy efficiency and environmental impacts of various vaccine options," *Energies (Basel)*, vol. 14, no. 7, pp. 1849–, 2021.

[6] I. Dinçer, *Refrigeration Systems and Applications*. Wiley, 2017.

[7] V. Stamer, H. Renz, and B. K. GmbH, *Grundlagen Der Kältetechnik: Fundamentals of Refrigeration*. tredition, 2021.

[8] M. Keir and A. Alleyne, "Dynamic model-based fault detection and diagnosis residual considerations for vapor compression systems," in *2006 American control conference*. IEEE, 2006, pp. 6–pp.

[9] Z. Soltani, K. K. Sørensen, J. Leth, and J. D. Bendtsen, "Fault detection and diagnosis in refrigeration systems using machine learning algorithms," *International Journal of Refrigeration*, vol. 144, pp. 34–45, 2022.

[10] Y.-J. Park, S.-K. S. Fan, and C.-Y. Hsu, "A review on fault detection and process diagnostics in industrial processes," *Processes*, vol. 8, no. 9, 2020.

[11] H. Habibi, I. Howard, and S. Simani, "Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review," *Renewable Energy*, vol. 135, pp. 877–896, 2019.

[12] M. M. Syed, T. A. Lemma, S. K. Vandrangi, and T. N. Ofei, "Recent developments in model-based fault detection and diagnostics of gas pipelines under transient conditions," *Journal of Natural Gas Science and Engineering*, vol. 83, p. 103550, 2020.

[13] Y. Li, "Exploring real-time fault detection of high-speed train traction motor based on machine learning and wavelet analysis," *Neural Computing and Applications*, vol. 34, no. 12, pp. 9301–9314, 2022.

[14] Y. Qin, Y. Yan, H. Ji, and Y. Wang, "Recursive correlative statistical analysis method with sliding windows for incipient fault detection," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 4185–4194, 2022.

[15] M. F. Tariq, A. Q. Khan, M. Abid, and G. Mustafa, "Data-driven robust fault detection and isolation of three-phase induction motor," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4707–4715, 2019.

[16] M. Quiñones-Grueiro, M. Ares Milián, M. Sánchez Rivero, A. J. Silva Neto, and O. Llanes-Santiago, "Robust leak localization in water distribution networks using computational intelligence," *Neurocomputing (Amsterdam)*, vol. 438, pp. 195–208, 2021.

[17] J. Sana Ullah, D. L. Young, and S. K. In, "A distributed sensor-fault detection and diagnosis framework using machine learning," *Information Sciences*, vol. 547, pp. 777–796, 2021.

[18] F. Guo, A. P. Rogers, and B. P. Rasmussen, "Multivariate fault detection for residential hvac systems using cloud-based thermostat data, part i: Methodology," *Science and Technology for the Built Environment*, vol. 28, no. 2, pp. 109–120, 2022.

[19] Z. Yang, X. Linda, and W. Shengwei, "An intelligent chiller fault detection and diagnosis methodology using bayesian belief network," *Energy and Buildings*, vol. 57, p. 278–288, 02 2013.

[20] K. Yan and X. Zhou, "Chiller faults detection and diagnosis with sensor network and adaptive 1d cnn," *Digital Communications and Networks*, 2022.

[21] W. Yao, D. Li, and L. Gao, "Fault detection and diagnosis using tree-based ensemble learning methods and multivariate control charts for centrifugal chillers," *Journal of Building Engineering*, vol. 51, p. 104243, 2022.

[22] Z. Yang, W. Shengwei, and X. Fu, "Pattern recognition-based chillers fault detection method using Support Vector Data Description (SVDD)," *Applied Energy*, vol. 112, no. C, pp. 1041–1048, 2013.

[23] K. Yan, J. Huang, W. Shen, and Z. Ji, "Unsupervised learning for fault detection and diagnosis of air handling units," *Energy and Buildings*, vol. 210, p. 109689, 2020.

[24] A. Montazeri and S. M. Kargar, "Fault detection and diagnosis in air handling using data-driven methods," *Journal of Building Engineering*, vol. 31, p. 101388, 2020.

[25] F. Guo, A. P. Rogers, and B. P. Rasmussen, "Multivariate fault detection for residential hvac systems using cloud-based thermostat data, part ii: Case studies," *Science and Technology for the Built Environment*, vol. 28, no. 2, pp. 121–136, 2022.

[26] S. Taheri, A. Ahmadi, B. Mohammadi-Ivatloo, and S. Asadi, "Fault detection diagnostic for hvac systems via deep learning algorithms," *Energy and Buildings*, vol. 250, p. 111275, 2021.

[27] Y. Guo, Z. Tan, H. Chen, G. Li, J. Wang, R. Huang, J. Liu, and T. Ahmad, "Deep learning-based fault diagnosis of variable refrigerant flow air-conditioning system for building energy saving," *Applied Energy*, vol. 225, pp. 732–745, 2018.

[28] W. Tun, J. K.-W. Wong, and S.-H. Ling, "Hybrid random forest and support vector machine modeling for hvac fault detection and diagnosis," *Sensors*, vol. 21, no. 24, 2021.

[29] K. Yan, Z. Ji, H. Lu, J. Huang, W. Shen, and Y. Xue, "Fast and accurate classification of time series data using extended elm: Applicationin fault diagnosis of air handling units," pp. 1349–1356, 2017.

[30] H. Zhang, C. Li, Q. Wei, and Y. Zhang, "Fault detection and diagnosis of the air handling unit via combining the feature sparse representation based dynamic sfa and the lstm network," *Energy and Buildings*, vol. 269, p. 112241, 2022.

[31] N. G. Lo, J.-M. Flaus, and O. Adrot, "Review of machine learning approaches in fault diagnosis applied to iot systems," in *2019 International Conference on Control, Automation and Diagnosis (ICCAD)*, 2019, pp. 1–6.

[32] J. Z. Y. Huang, "Ga and rbf based real-time fdd for refrigeration units," in *2009 International Symposium on Intelligent Ubiquitous Computing and Education*, May 2009, pp. 22–25.

[33] K. Manske, D. Reindl, and S. Klein, "Evaporative condenser control in industrial refrigeration systems," *International Journal of Refrigeration*, vol. 24, no. 7, pp. 676 – 691, 2001.

[34] S. Tassou and I. Grace, "Fault diagnosis and refrigerant leak detection in vapour compression refrigeration systems," *International Journal of Refrigeration*, vol. 28, no. 5, pp. 680 – 688, 2005.

[35] A. Behfar, D. Yuill, and Y. Yu, "Automated fault detection and diagnosis methods for supermarket equipment (rp-1615)," *Science and Technology for the Built Environment*, vol. 23, no. 8, pp. 1253–1266, 2017.

[36] 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.

[37] A. Vogelsang and M. Borg, "Requirements engineering for machine learning: Perspectives from data scientists," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 2019, pp. 245–251.

[38] M. W. Berry, A. Mohamed, and B. W. Yap, *Supervised and unsupervised learning for data science*. Springer, 2019.

[39] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[40] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[41] T. Mahmood, J. Li, Y. Pei, F. Akhtar, S. A. Butt, A. Ditta, and S. Qureshi, "An intelligent fault detection approach based on reinforcement learning system in wireless sensor network," *The Journal of Supercomputing*, vol. 78, no. 3, pp. 3646–3675, 2022.

[42] Z. Soltani, K. K. Soerensen, J. Leth, and J. D. Bendtsen, "Fault detection of supermarket refrigeration systems using convolutional neural network," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 231–238.

[43] J. Yang and G. Yang, "Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer," *Algorithms*, vol. 11, no. 3, 2018.

[44] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[45] R. Rodríguez-Pérez and J. Bajorath, "Evolution of support vector machine and regression modeling in chemoinformatics and drug discovery," *Journal of Computer-Aided Molecular Design*, pp. 1–8, 2022.

[46] V. Vapnik, *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

[47] M. N. Murty and R. Raghava, *Support Vector Machines and Perceptrons: Learning, Optimization, Classification, and Application to Social Networks*, ser. SpringerBriefs in computer science. Springer International Publishing AG, 2016.

[48] Z. Soltani, K. K. Sørensen, J. Leth, and J. D. Bendtsen, "Robustness analysis of pca-svm model used for fault detection in supermarket refrigeration systems," in *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE, 2021, pp. 1–6.

[49] Z. Yang, K. B. Rasmussen, A. T. Kieu, and R. Izadi-Zamanabadi, "Fault detection and isolation for a supermarket refrigeration system – part one: Kalman-filter-based methods," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13 233 – 13 238, 2011, 18th IFAC World Congress.

[50] R. N. Brown, "3 - reciprocating compressors," in *Compressors (Third Edition)*, R. N. Brown, Ed. Gulf Professional Publishing, 2005, pp. 52–124.

[51] H. Farhat, "Chapter 4 - maintenance: Availability and reliability," in *Operation, Maintenance, and Repair of Land-Based Gas Turbines*, H. Farhat, Ed. Elsevier, 2021, pp. 89–106.

Part II

Papers

Paper A

Fault Detection of Supermarket Refrigeration Systems Using Convolutional Neural

Zahra Soltani, Kresten Kæjr Sørensen, John Leth, Jan Dimon Bendtsen

The paper has been published in the
IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society,
pp. 231–238, 2020.

The layout has been revised.

Abstract

The functionality of supermarket refrigeration systems (SRS) has a significant impact on the quality of food products and potentially human health. Automatic fault detection and diagnosis of SRS is desired by manufacturers and customers as performance is improved, and energy consumption and cost is lowered. In this work, Convolutional Neural Networks (CNN) are applied for fault detection and diagnosis of SRS. The network is found to be able to classify the fault with 99% accuracy. The sensitivity of the designed model to data quality is also assessed. The results show that the model can classify faults at low sample rates if the training set is large enough. Moreover, the model displays low sensitivity to data quality such as noisy and perturbed validation data, and the frequency of false positives is satisfactorily low as well.

1 Introduction

The general quality of refrigerated food depends on how accurately its temperature is controlled throughout the cold-chain, from production to the end-users. Improvement of reliability of Supermarket Refrigeration Systems (SRS) by early fault diagnosis is highly relevant when considering the safety of food, human health, and environmental pollution of a large industry. According to [1], because SRS must run night and day, they consume about 50% of the entire energy budget of most supermarkets. Thus, using a faulty refrigeration system can lead to critical economic losses. As a consequence, refrigeration companies try to gain a competitive edge by producing products with as high degree of automation as possible, including for performance monitoring or fault diagnosis.

In this paper, classification of evaporator fan faults is studied; these faults may typically result in inaccurate cooling room temperature, which may lead to food spoilage and energy waste. Therefore, it is of high importance to detect and diagnose evaporator fan faults before they result in damage to the goods. However, constant human monitoring is tedious, expensive and error-prone. Therefore, data-driven Fault Detection and Diagnosis (FDD) has become increasingly popular in the industry, and in particular Artificial Intelligence (AI) is receiving a lot of attention due to its abilities to make decision instantaneously and deal with vast amounts of data [2].

Convolutional Neural Networks (CNN) is a cornerstone in image processing when it comes to the classification of highly challenging data sets [3]; CNNs are known to be accurate and computationally faster than most other machine learning-based classification methods. In this method, the essential features, information or correlation among data is extracted. Afterwards, the data are classified based on the information. Similar ideas can be used in the classification of signals in signal processing. From this view, CNNs can be used for fault detection and classification.

A number of different data-driven approaches have been proposed. For instance, a combination of a Genetic Algorithm and a Pseudo-inverse matrix algorithm can be found in [4] to obtain parameters and weights of the radial basis function network. This method identifies, successfully, six faults emulated in a laboratory set up. An Extended Kalman Filter (EKF) method is proposed in [5]. Although the EKF performed better and faster fault detection than an ordinary Kalman filter, neither filter was able to distinguish between sensor faults and parametric faults. On the other hand, successful results of Artificial Neural Network (ANN) strategies regarding FDD in chillers have been reported, e.g., [6], and [7]. The Support Vector Data Description (SVDD) method was employed for FDD in chillers in [7], where SVDD is compared with the Principal Component Analysis (PCA) method. The fault detection performance of the SVDD method was better than the PCA method. In [8], refrigerant leaks were detected and diagnosed using a probabilistic ANN algorithm; the ANN could detect refrigerant losses with 90% accuracy. In [9], the better performance of a Probabilistic Neural Network is demonstrated compared to Back Propagation (BP) method as BP has random initial wights leading to a less reliable system. Air handling unit faults are detected using a pattern matching method in [10]. This method is combined with PCA in [11]. It improves the sensitivity of fault detection model and boosts the performance of air handling units.

In image recognition applications CNN is known because of its impressive feature extraction and classification capabilities. These capabilities make it a strong candidate for FDD and process monitoring, where fault patterns might appear in data without being immediately apparent to human observers, see [12].

This research contribution is a CNN model for fault identification and sensitivity analysis of the model to the data quality. The model can classify a specific fault on the evaporation side of a refrigeration system, using only indirect measurements gathered from the condensing side of the system. The structure of available data can vary in the field due to different requirements and configurations of SRS. For instance, the sample rate when acquiring data can be between 1 to 0.0003 Hz; the number of samples or length of the data logs varies depending on embedded hardware type and software requirements. To some extent, there would be different correlations between data parameters due to variations in SRS components and loads. Thus, the sensitivity of the model against the variation of data structure is tested and improved. The model is found to be able to classify validation data with 99% accuracy, and exhibits roughly low sensitivity to low-resolution, noisy, and perturbed data. Non-faulty perturbed data are classified with the same accuracy, but with less reliability (99% accuracy in 92 of 100 trials).

The outline of the rest of the paper is as follows: Section 2 explains SRS preliminaries, the general methodology of CNN, and its training process. In section 3, data collection and CNN design is represented. Different sensitivity tests against data quality are proposed in section 4, and the results are investigated in section 5. Finally, conclusions are presented in sections 6.

2 Preliminaries

2.1 Supermarket refrigeration systems

Refrigeration systems transfer heat in a process where heat is absorbed from a cooling room and released in the ambient environment. During a cycle, the heating absorption and heat dissipation happen by changing the refrigerant phase from liquid to vapour and vapour to liquid, respectively. The nomenclature of the SRS and refrigeration shown in Fig. A.1, are introduced in Table A.1. This system includes two parts; a Bitzer condensing unit and an industrial evaporator for air cooling mounted in an insulated room. The number of evaporators, fans and cooling rooms are different from supermarket to supermarket. In Fig. A.1, $Ctrl_{\text{Cond}}$ is the condensing unit controller which is connected to the required sensors to control the condenser fan speed and compressor speed V_{cpr}. The compressor speed is controlled to provide the capacity required to keep the temperature in the cooling room. Inside the cooling room, heat is transferred from the goods via the evaporation process to the refrigerant. The evaporator controller is called $Ctrl_{\text{Evap}}$ and can be seen in Fig. A.1. This controller uses required inputs taken from sensors to control the evaporator fan speed and opening degree of the expansion valve. The opening degree of the expansion valve determines the amount of refrigerant that passes through the evaporator and is controlled to achieve Minimum Stable Superheat (MSS).

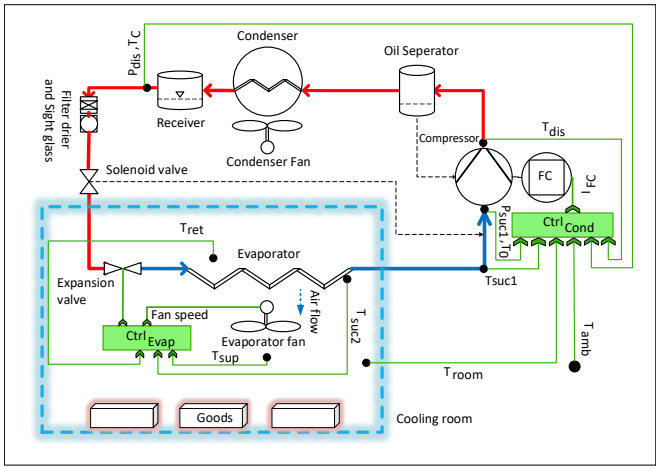

Fig. A.1: Schematic of a refrigeration system.

The evaporator fan is responsible for circulating air over the evaporator surface and in the cold room to enhance heat transfer into the evaporator and keep an even

Table A.1: Symbols used in the Fig. A.1

Symbols	description	SI unit
T_{room}	cooling room temperature (sensor)	$[°C]$
T_{amb}	ambient temperature (sensor)	$[°C]$
$T_{suc1,2}$	suction temperature (sensor)	$[°C]$
T_0	evaporation temperature (sensor)	$[°C]$
P_{suc}	suction pressure (sensor)	$[Pa]$
T_{dis}	discharge temperature (sensor)	$[°C]$
$Pdis$	discharge pressure (sensor)	$[Pa]$
T_c	condensing temperature	$[°C]$
T_{ret}	returned air temperature (Sensor)	$[°C]$
T_{sup}	supplied air temperature (Sensor)	$[°C]$
I_{FC}	converter current	$[A]$
FC	frequency converter	[-]
$Ctrl_{Evap}$	evaporator controller	[-]
$Ctrl_{Cond}$	condensor controller	[-]

temperature in the cold room. When the fan does not work or run slowly, there would not be enough airflow around the evaporator pipes. The reduced airflow causes reduced heat transfer. Thus, to compensate and keep the required cooling capacity for the room, $Ctrl_{\text{Evap}}$ increases the temperature difference between the refrigerant and the air. This causes the suction pressure P_{suc} to drop and the vapor density ρ_v at the compressor inlet to decrease as:

$$\rho_v = \frac{P_{\text{suc}} M_{\text{m}}}{R T_0} \tag{A.1}$$

where M_{m} is Molar mass and T_0 is evaporation temperature of the refrigerant and R is the ideal refrigerant gas constant. Lower ρ_v leading to more work required by the compressor and increased mass flow rate $\dot{m}$ as:

$$\dot{m} = \rho_v V A \tag{A.2}$$

where V is the volumetric flow rate of the refrigerant and A is the area of the compressor inlet. Therefore, $\dot{m}$ could be one of the indicators for a faulty evaporator fan. However, it requires knowledge of the compressors parameters, which is not always available. Therefore, the proportion to the mass flow rate is enough, where A is omitted and the compressor speed (V_{cpr}) is used instead of V. Note that V_{cpr} is proportional to V itself. Thus, a proportional compressor mass flow rate K_{mf} can be used, represented as:

$$K_{\text{mf}} \propto \rho_v V_{\text{cpr}}. \tag{A.3}$$

In the beginning stage, the evaporator fan fault leads to higher compressor speed and more power consumption. If the compressor reaches its maximum speed due to excessively low P_{suc} the temperature in the cold room will begin to increase. This implies violation of the food quality. Therefore, early FDD is required before any change in the cooling room temperature occurs.

2.2 CNN methodology

In the sequel, the CNN methodology adopted for fault detection will be presented.

Suppose we are given a set of feature vectors $\{\chi_k\}, k = 1, \ldots, K$, each of which belong to a finite set of classes $\{\kappa_n\}, n = 1, \ldots, \nu$. The associated *classification problem* is then the challenge of finding a map $\mathcal{N} : \mathcal{X} \to \{e_n\}, n = 1, \ldots, \nu$, where $\mathcal{X}$ is the *feature space* from which χ_k are drawn and $\{e_n\}$ is an orthonormal set of vectors with all entries equal to zero except the j'th entry, which is one; e_n corresponds to class n.

The map $\mathcal{N}$ will be approximated using a CNN. CNNs are composed of *neurons*, which are nonlinear functions parametrized by so-called *synaptic weights*. The neurons are organized in *layers* – an input layer, several hidden layers and an output layer – and trained using supervised learning. Commonly, CNNs can be decomposed into two separate stages. The first stage, called the feature extraction stage, includes the input layer along with one or more convolutional layers. The second stage includes a number of fully connected layers, which are responsible for the classification – see Fig. A.2. The most informative features are collected in the last convolution layer. A flattening layer is the vectorized shape of the last convolutional layer used as an input to the classification stage. The number of neurons of the output layer should match the number of classes ν.

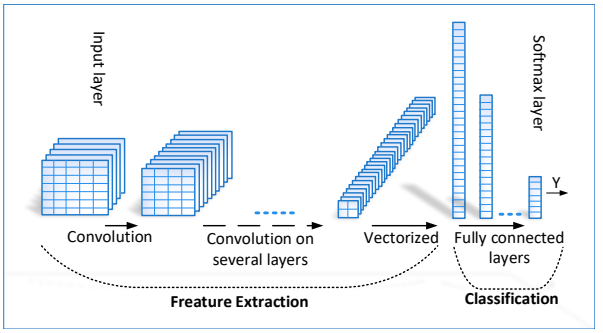

Fig. A.2: General design of CNN.

2.3 Training the model

The first stage of a CNN is organized like a standard multi-layer perceptron network, i.e., all nodes in each layer are feed-forward connected by weights $w_i \in \mathbb{R}$ from inputs $x_i \in \mathbb{R}, i = 1, \ldots, n$ via a neuron function $f : \mathbb{R} \to \mathbb{R}$ to yield a neuron output y:

$$y = f\left(\sum_{i=1}^{m} w_i x_i - b\right). \tag{A.4}$$

The input to the first layer is χ_k. The output y from each layer is subsequently used as input for all neurons in the subsequent hidden layer. In neural network terminology, b is called the activation threshold or bias, while the sum of weighted inputs and bias is known as an activation potential, and m is the number of neurons in previous layer. A layer is thus a column vector of neurons, each of which may be parametrized by different sets of weights.

In a CNN, the last layer $F : \mathbb{R}^\nu \to \mathbb{R}^\nu$ is often chosen as a so-called *Softmax* activation function, whose n'th component is defined as:

$$\hat{Y}_{k,n} = F_n(y_{k-1,n}) = \frac{\exp(y_{k-1,n})}{\sum_{n=1}^{\nu} \exp(y_{k-1,n})} \tag{A.5}$$

where $\hat{Y}_k = \hat{Y}_k(W^p)$ is the CNN's estimate of the class of the k'th feature vector based on the current set of weights W^p. The Softmax function is a smooth approximation to the function $\arg\max(\cdot)$, basically picking out the index n among the entries of the input y_{k-1} with the largest value; that is, $\hat{Y}_k \approx e_n$ if the largest entry in y_{k-1} is found at index n.

To begin training a CNN, it is first necessary to obtain the training data and desired output corresponding to each input. For each class the training data are divided into specific size and each small part is called a mini-batch. The mini-batches stack together at the third dimension. Fig. A.3 illustrates the input pre-processing required for training the CNN and the output structure. The CNN needs to learn the corresponding desired output for each input. In addition, for all of the mini-batches in the same class, corresponding outputs are the same and the number of Training mini-batches in each class (T_{mb}) is:

$$T_{\mathrm{mb}} = n_{\mathrm{d}} \left\lfloor R \left\lfloor \frac{N_{\mathrm{s}}}{S_{\mathrm{mb}}} \right\rfloor \right\rfloor \tag{A.6}$$

where n_{d} is number of data logs, R is split ratio between training and test data, N_{s} is the number of samples, S_{mb} is the size of each mini-batch. After designing the shape of each class of functional and faulty system, The inputs to the CNN require both classes of functional and faulty data. Thus, both classes are concatenated, as shown in Fig. A.4. The number of output neurons in the CNN is the same as the number of classes. *Training* is a process in which the network weights are updated to give

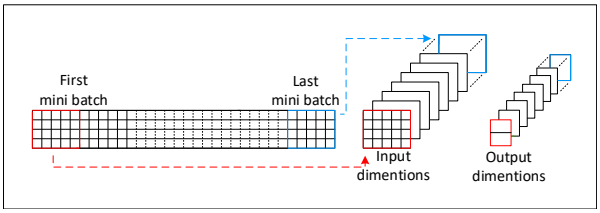

Fig. A.3: Input pre-processing and the output structure of each class.

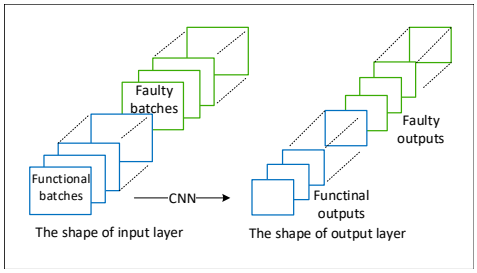

Fig. A.4: Visualisation of input and output shape after concatenation of non-faulty and faulty mini-batches.

increasingly better predictions of the correct classes as a function of the input feature vectors. Each update of the weights is called an *epoch*. The improvement in prediction is measured by way of a *Loss function*, which should be selected to match the activation function of the output layer; the *cross-entropy* loss function is commonly chosen in classification tasks (as opposed to, for example, the sum-of-squared-error loss function used in function approximation). When only two classes are considered, one may choose a *sigmoid* neuron in the output layer, which always yields an output prediction between zero and one, which may, in turn, be interpreted as a probability of the given feature belonging to the corresponding class. Training with the cross-entropy as the loss function then corresponds to maximizing the conditional log-likelihood of the data being correctly classified as explained in [13].

Given a collection of network weights W^p and ν independent targets (classes), the cross-entropy error for a single example χ_k is given by

$$E_k(\hat{Y}_k, W^p) = -Y_k^\top \ln\left(\hat{Y}_k\right) - (\mathbf{1} - Y_k)^\top \ln\left(\mathbf{1} - \hat{Y}_k\right). \tag{A.7}$$

where $\mathbf{1} = [1, 1, \ldots, 1]^\top$ and $\ln(\cdot)$ is taken element-wise to yield a ν-dimensional output.

This function estimates the difference between the actual and predicted probability distribution. *Stochastic Gradient Descent* (SGD) optimization is used to tune the

weights to improve the prediction–see [14]. The weights in layer l are adjusted in epoch p using

$$w_{il}^{(p)} = w_{il}^{(p-1)} + \alpha \nabla E_k(\hat{Y}_k, W^p) \tag{A.8}$$

where α is *learning rate* or step size and $\nabla E_k(\hat{Y}_k, W^p)$ is the gradient of the loss function wrt. the weights. We may compute the derivative of the cross-entropy error with respect to each weight connecting the hidden layer neurons to the output layer neurons using the chain rule:

$$\frac{\partial E_k}{\partial w_{n,i}} = \frac{\partial E_k}{\partial \hat{Y}_{k,n}} \frac{\partial \hat{Y}_{k,n}}{\partial u_n} \frac{\partial u_n}{\partial w_{n,i}}$$

where $u_n = \sum_i w_{n,i} x_i - b_n$ is the input to the n'th neuron in the previous layer. In each epoch, the calculated loss is propagated backward in the network in a layer-by-layer sequential fashion, where the gradients are computed from (A.4). *Adaptive Moment estimation* (*Adam*) is a variation of SGD, in which the learning rate α is tuned adaptively to deal with sparse gradient and non-stationary objectives. Moreover, the Adam optimizer is capable of dealing with falling into local minima; see [15] for details.

Convolution of the filters or weights through the feed-forward process prevents having a vast number of weight vector connections in every layer and speed up the network operation. Besides, non-informative features can be eliminated in each layer using the so-called *pooling method*. By using pooling after the convolutional layer, the outputs of each layer are pooled together in the specified filter size. The most common pooling methods are average-pooling, and max-pooling, which collect the average of the outputs, and a maximum of the outputs, respectively.

3 Experiments

The condensing unit in the laboratory shown in Fig. A.1, consists of a semi-hermetic reciprocating, four-cylinder compressor with a speed range of 25-87 Hz. It has 17KW cooling capacity at 10 ^{o}C evaporating temperature using refrigerant R-134a. The two condenser fans have a maximum power consumption of 350 W.

Supermarket condensing units are connected to different evaporation setups, depending on the requirements of each supermarket. Moreover, information from the evaporation part of refrigeration systems is typically not available. In this paper, data is taken from the condensing unit, and the data from the evaporation side is neglected. The evaporator at the Bitzer electronic laboratory has two fans. Fan speeds are controlled by $\text{Ctrl}_{\text{Evap}}$ shown in Fig. A.1. In order to emulate the evaporator fan fault, a switch is installed between controller output and relay of one of the fans as seen in Fig. A.5. Thus, it is possible to switch on and off, manually, one of the fans and collect data in both conditions.

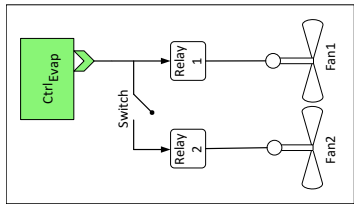

Fig. A.5: Stop force on the evaporator fan to emulated defective fan.

3.1 Data acquisition

Data is taken where the room setpoint varies in the range of 1 ° C to 12 ° C. Different compressor speeds between 33Hz and 70Hz are applied. This variation is needed in the training phase to learn the response of the system in different states. Moreover, it prevents the over-fitting of the neural network.

When one of the fans is switched off the defective fan fault is emulated on the laboratory set up. Both non-faulty and faulty data are collected with 1 Hz sample rate. Each set of data includes information of P_{suc}, Superheat temperature T_{sh}, V_{cpr} and K_{mf}. These parameters change when the fan is switched off while room temperature remains constant and controlled. When the fault occurs, V_{cpr} increases to compensate P_{suc} and density drop at the inlet of the compressor. This means that the overall efficiency of the system is reduced due to the fault, but, because the system is able to keep the room temperature, this fault would normally not be detected by traditional fault detection. Moreover, K_{mf} is oscillating more because there is less ventilation around the evaporator, and it causes unstable heat transfer around the evaporator. This oscillatory heat transfer induces oscillations in T_{suc} and P_{suc}, which in turn results in K_{mf} oscillating with a higher amplitude. While this fault continues in the system, the cooling room temperature at different locations varies. Finally, due to the lower P_{suc}, and reduced heat transfer, the evaporator surface is colder, and this leads to a faster build-up of ice on the evaporator. Therefore, detecting of accumulated ice is needed more regularly, and this also presents an additional energy cost.

Therefore, a CNN algorithm is used to detect the fault before it affects the room temperature and prevents excessive energy usage due to inefficient running without the fault detection.

3.2 CNN specification

In this work, six data logs corresponding to various loads and set-points are used. The size of each data log is 4×13000 samples because there are four measurements in each data set, as mentioned in Subsection 3.1. When designing CNNs, it is important to

select proper hyper-parameters. Hyper-parameters are external and controllable parameters set by the user, including mini-batch size, number of layers, activation functions, filter size, cost function, and optimization method, and so on. In this paper, the mini-batch size is selected as 4×30 samples, which is obtained by manual optimization. The initial learning rate is a key parameter in the training configuration; here α in (A.8) is chosen as 0.0003. At the output layer a Sigmoid function

$$\hat{Y}_1 = \frac{1}{1 + \exp(y_{k-1})} \tag{A.9}$$

is used because it is a binary classification. The range of the Sigmoid function is $[0, 1]$, and the classification is performed by a simple threshold; if $\hat{Y}_1 < 0.5$, the class is 0, and 1 otherwise.

In this work, *binary-cross-entropy*

$$E(\hat{Y}_1, W^p) = -Y_1 \ln\left(\hat{Y}_1\right) - (1 - Y_1) \ln\left(1 - \hat{Y}_1\right) \tag{A.10}$$

is used as a cost function, which is the same as the cross entropy in (A.7) for only two output vectors. In (A.10), $Y_1 = 1$ is the value assigned to class one and $\hat{Y}_1 \in [0, 1]$ is the estimated probability of the input sample belonging to that class. Since probabilities sum to 1, the second class is assigned the value $Y_2 = 1 - Y_1$ and the corresponding estimate is $\hat{Y}_2 = 1 - \hat{Y}_1$.

In this work, the design of the CNN is improved as table A.2 to obtain better classification results. In this table, S_f stands for the size of the filters, N_f is the number of filters in each layer, Act is activation function where ReLU stands for Rectifier Linear Unit, and MP is Maxpooling size. Padding type is mentioned as P and valid means that an array of zeroes is applied to the edges of the data when passing through the next layer. The fully connected layer is used with 50% dropout. *Dropout* is an efficient

Table A.2: Design of the CNN algorithm, used in this work.

Layer	S_f	N_f	Act	P	MP
convolution	(2,20)	16	ReLU	valid	(1,3)
convolution	(2,3)	32	ReLU	valid	-
convolution	(1,3)	64	ReLU	valid	-
Flatten	-	-	-	-	-
FC	40	-	ReLU	-	-
Dropout(0.5)	-	-	-	-	-
FC	2	-	Sigmoid	-	-

solution to prevent over-fitting. In this method, the number of neurons is regularized

in each layer, and the rest of the neurons are dropped temporarily together with all the inputs and outgoing connections [16].

4 Sensitivity to data quality

As SRS operations and configurations vary significantly due to the individual supermarket's demands and geographical conditions, it is very important that the FDD model has low sensitivity to variations in the data available. Here, a number of experiments is introduced to examine the sensitivity to data quality and reliability of the model. For all experiments, the structure of the model and the hyper-parameters of the CNN model are the same as specified in the table A.2.

4.1 Low resolution data

In SRS data acquisition, the sample rate varies between 1 to 0.0003 Hz depends on the embedded hardware memory. In this work, data is re-sampled from 1 Hz to 0.16, 0.016, and 0.0016 Hz. Even though it was of interest to observe the result with much lower sample rates such as 0.0003 Hz, it is not possible to do this here, due to limitations on data log length. One non-faulty and one faulty data log with regards to four different sample rates are introduced in Fig. A.6. From top to bottom, the sample rate is decreased by means of down-sampling; these re-sampled data sets will be used for training CNN models. As can be seen, lowering the sample rates from 1 to 0.016 Hz does not change the main features of the data due to the slow dynamics of the refrigeration system. However, at 0.0016 Hz the important features can no longer be detected in the sampled data.

As the data is down-sampled, the number of samples decreases. Therefore, another experiment is done to train the model with the same data length to compare the down-sampling result with the same size of data. The results of these experiments are presented in subsection 5.1.

4.2 Noisy data

In industrial applications, data can be noisy and incomplete due to different reasons, for instance, sensor noise, electromagnetic propagation, sample dropouts and so forth. In this work, noise is added to data, not only to observe noisy data response but also to generate random new validation data where correlation among parameters is preserved. Random variables with normal distribution $\mathcal{N}(0, 2)$ is selected for this purpose. To do this, the noise S_{n} is added to the saturation temperature T_{sat} and consequently to T_{sh} as:

$$T_{\mathrm{sh}} + S_{\mathrm{n}} = (T_{\mathrm{suc}} - T_{\mathrm{sat}}) + S_{\mathrm{n}} \tag{A.11}$$

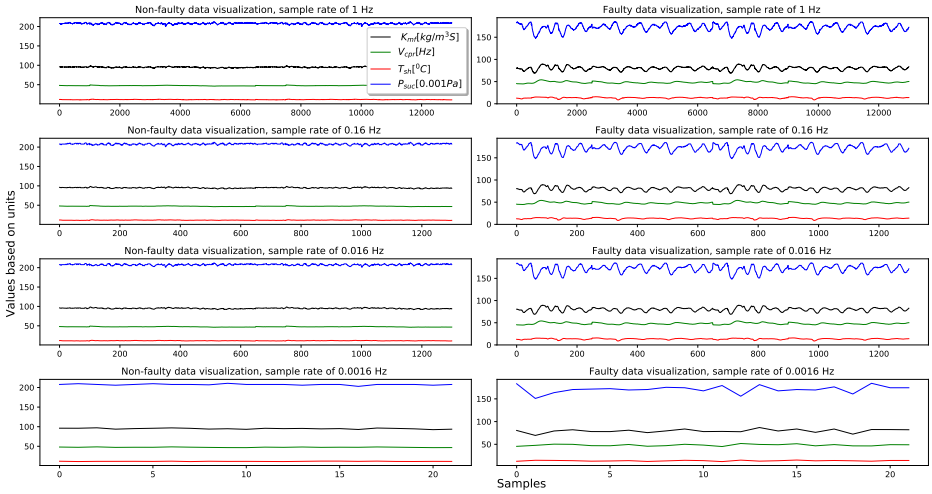

Fig. A.6: Visualisation of data re-sampling for Non-faulty data to the left and faulty data to the right.

where T_{suc} is suction temperature or actual temperature of refrigerant after evaporation. Therefore, by any change in T_{sh}, P_{suc} and eventually K_{mf} would change as the correlations are introduced in (A.1) – (A.3).
In Fig. A.7, Random noise with $\mathcal{N}(0, 2)$ is added to T_{sh}, and eventually, P_{suc} and K_{mf}. In order to test the reliability of the algorithm, noisy validation data is generated 100 times and passed through the network as new validation data sets. The result of this test can be found in subsection 5.2.

4.3 Operating point change

The SRS may operate in different operating points depending on the needed capacity, the layout of the system and the ambient conditions. To the CNN, this will look like offsets or perturbations in measurements that are correlated according to the physics of the system. Thus, a random offset value is applied to T_{sh} on validation data set using random numbers between [-3,3.5]. In accordance with (A.1) – (A.3), P_{suc} and K_{mf} change correspondingly; as P_{suc} should be in its valid refrigeration cycle envelope, the random offset value can not be outside specific ranges when using the available data. This random perturbation is applied 100 times to observe the reliability of the model when the correlation between parameters is different from what is used in the training data.

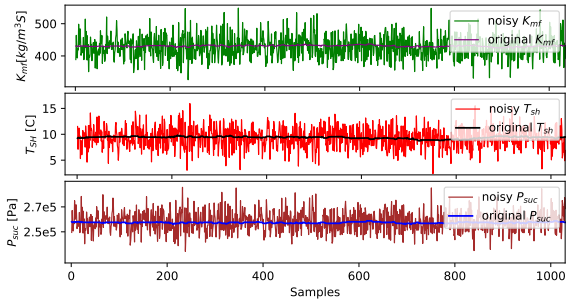

Fig. A.7: Example of original and noisy data where noise is added to T_{sh}.

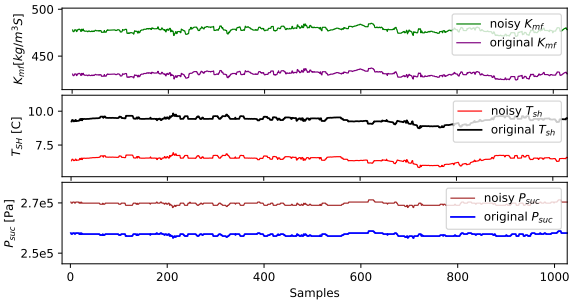

Fig. A.8: Example of original and perturbed data.

5 Results

A number of different experiments to evaluate the sensitivity of the CNN model were proposed in the previous section. The results of each experiment are presented in the following.

5.1 Data re-sampling result

Re-sampling of the training data is done as specified in subsection 4.1. Fig. A.6 illustrates that even when the sampling rate is reduced and the number of samples is lowered, the main features in the data are preserved due to the slow dynamic of SRS. Fig. A.9 shows the accuracy of classification and loss function with different sample rates. It is seen that lowering the sample rate also lowers the accuracy of the training process and causes the training to require more iterations. However, it should be noted that the decreased accuracy is not so much due to the lower sample rate itself, but rather due to the lower number of samples available to the CNN. In Fig. A.6, down-sampling

is continued until 0.0016 Hz in order to obtain a lower bound on the sampling rate; however, with the few data points remaining, it becomes impossible to train the CNN at this sample rate.

Fig. A.9 indicates that if the sample rate is kept constant, the accuracy is decreased when the data is shortened (see the experiment in blue, orange, red and brown). It is remarked that the experiments in orange and green (respectively red and purple) have the same data length, but the green (respectively purple) with a lower sample rate has faster convergence. Note that for each of the data lengths, the figure shows the lowest sampling rates for which training was successful.

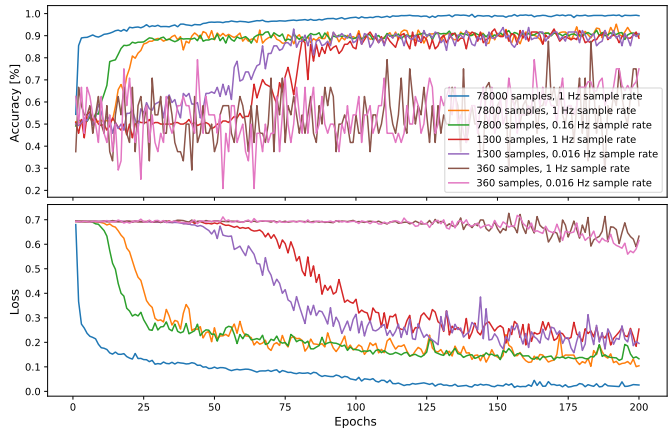

Fig. A.9: Resampling evaluation when the number of samples are constant.

Fig. A.10 presents zoomed-in data with 1 and 0.16 Hz sample rates. It appears that using short mini-batches with relatively high sample rates causes the oscillations observed in the faulty data to disappear (the data is near-constant over these periods), and that oscillations are important features of the faulty data. Thus, the reason that the green (purple) result is better than the orange (red) one in Fig. A.9 is that the low-resolution faulty mini-batches are easier to classify than high-resolution mini-batches.

5.2 Noisy data result

In Subsection 4.2, the method of generating validation data with specified noise is explained. The result of 100 stochastic tests over the CNN model is introduced in Fig. A.11. This figure shows how accurately 100 faulty data-sets and 100 non-faulty data-sets are classified. The value at the top of each column shows the distribution of corresponding accuracy values when running 100 stochastic tests. Non-faulty data is

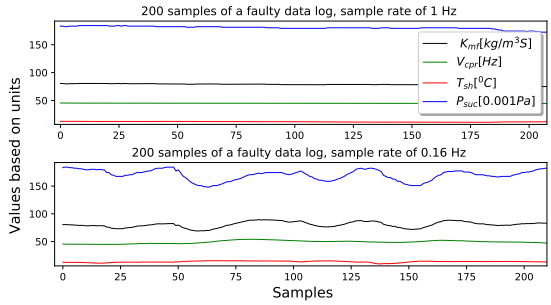

Fig. A.10: Two zoomed data logs with different sample rates.

classified with higher than 99% accuracy with 100% reliability. Faulty data is classified with better than 97% accuracy in 95 out of the 100 runs.

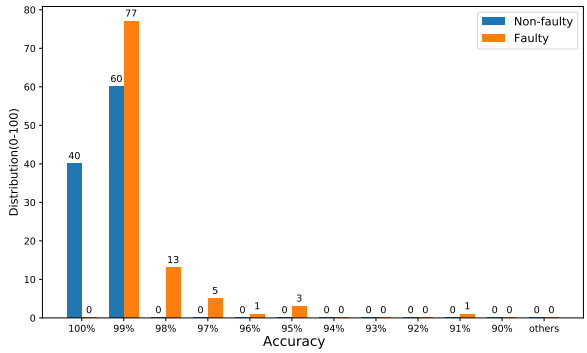

Fig. A.11: Distribution of classification accuracy achieved when running the CNN algorithm 100 times using noisy validation data.

5.3 Result of operating point change

As explained in Subsection 4.3, due to different SRS configurations and loads in the cold room, the data can be varied while the correlation among parameters is preserved. The result of 100 runs is represented in Fig. A.12. Note that perturbation of the parameters is limited as explained in Subsection 4.3. The model has good classification capabilities; for the faulty data, 99% accuracy is achieved in all of the 100 runs. On the other hand, non-faulty data are detected correctly with 99% accuracy in 92 out of the 100 runs, while in 8% of the runs non-faulty data was classified with less than 91% accuracy.

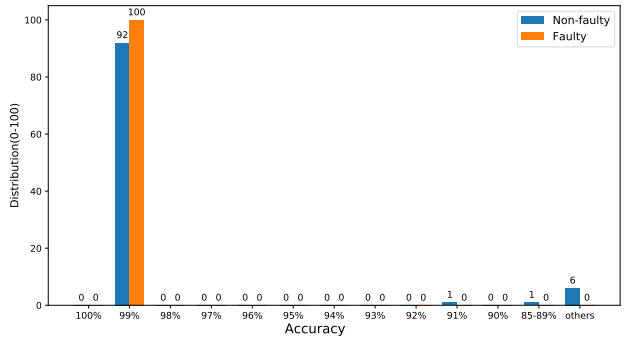

Fig. A.12: Distribution of classification accuracy when running the CNN algorithm 100 times using perturbed validation data.

5.4 False positive analysis

A model is said to give a false positive classification when it incorrectly indicates that a system is faulty while it is, in fact, healthy. False positives must occur as rarely as possible, because it results in unnecessary costs for supermarket owners for changing components or doing inspection. As shown in Fig. A.11, 1% false positives were classified in 60 out of the 100 runs using noisy data, while there were no false positives in the remaining 40 runs. Moreover, stochastic perturbation of the data yielded 1% false positives for 92 out of 100 runs, see Fig. A.12. This experiment shows more than 9% false positives in 8 out of the 100 runs.

6 Conclusion

In industrial applications, diagnosis of a defective evaporator fan is not always timely, because the inspection is only done when the cooling room temperature exceeds its allowable range. In this paper, a CNN model is applied to detect an evaporator fan fault while the room temperature is actively controlled. Only data from the condensing unit was used because data of the evaporation side is not always available. An evaporator fan fault was emulated on a laboratory SRS, and the data was used to train and analyze the sensitivity of the CNN model to the data quality.

Fast sampling is expensive and monitoring is tedious, therefore one cannot normally expect data of the high quality shown in Fig 6 to be available during normal operation. It was therefore necessary to examine lower sampling rates and shorter data log lengths in order to assess practical classification scenarios.

It was found that using short mini-batches with relatively high sample rates causes the oscillations observed in the faulty data to disappear (the data is near-constant over

these periods), and that oscillations are important features of the faulty data.

Moreover, the sensitivity of the model against noisy validation data was studied as well. The noisy and faulty data were classified with better than 98% accuracy for 90 runs out of 100. Maximum 1% false positive classification was achieved when using noisy data.

Validation data acquired at different operating points were classified as well. In these cases, faulty data were classified with 99% accuracy for all 100 runs. For 92 runs out of 100, only 1% false positive classification was observed, which is a satisfactory result from a practical point of view. It is believed that the higher false positive classification (8% of the runs) can be improved if other random perturbed data is used during the training process. This method can be further developed to classify a number of different faults in SRS systems, allowing automatic early detection of costly faults, which human operators are unlikely to spot during day-to-day operation. Detecting potential faults prevent unnecessary fatigue, leading to lower economic losses to the operator/owner of the system.

References

[1] J. Arias, *Energy usage in supermarkets - modelling and field measurements*. Stockholm: KTH Industrial engineering and management, 2005.

[2] C. Ji, Y. Li, W. Qiu, U. Awada, and K. Li, "Big data processing in cloud computing environments," in *2012 12th International Symposium on Pervasive Systems, Algorithms and Networks*, Dec 2012, pp. 17–23.

[3] A. Krizhevsky and K. A, "Imagenet classification with deep convolutional neural networks," *NIPS*, pp. 1097–1105, 2012.

[4] J. Z. Y. Huang, "Ga and rbf based real-time fdd for refrigeration units," in *2009 International Symposium on Intelligent Ubiquitous Computing and Education*, May 2009, pp. 22–25.

[5] Z. Yang, K. B. Rasmussen, A. T. Kieu, and R. Izadi-Zamanabadi, "Fault detection and isolation for a supermarket refrigeration system – part one: Kalman-filter-based methods," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13 233 – 13 238, 2011, 18th IFAC World Congress.

[6] Z. Yang, X. Linda, and W. Shengwei, "An intelligent chiller fault detection and diagnosis methodology using bayesian belief network," *Energy and Buildings*, vol. 57, p. 278–288, 02 2013.

[7] Z. Yang, W. Shengwei, and X. Fu, "Pattern recognition-based chillers fault detection method using Support Vector Data Description (SVDD)," *Applied Energy*, vol. 112, no. C, pp. 1041–1048, 2013.

[8] K. Assawamartbunlue and M. J. Brandemuehl, "Refrigerant leakage detection and diagnosis for a distributed refrigeration system," *HVAC&R Research*, vol. 12, no. 3, pp. 389–405, 2006.

[9] Q. Liang, H. Han, X. Cui, H. Qing, and Y. Fan, "Comparative study of probabilistic neural network and back propagation network for fault diagnosis of refrigeration systems," *Science and Technology for the Built Environment*, vol. 24, no. 4, pp. 448–457, 2018.

[10] M. Najafi, D. M. Auslander, P. L. Bartlett, P. Haves, and M. D. Sohn, "Application of machine learning in the fault diagnostics of air handling units," *Applied Energy*, vol. 96, pp. 347 – 358, 2012, smart Grids.

[11] S. Li and J. Wen, "Application of pattern matching method for detecting faults in air handling unit system," *Automation in Construction*, vol. 43, pp. 49 – 58, 2014.

[12] Y. H. Eom, J. W. Yoo, S. B. Hong, and M. S. Kim, "Refrigerant charge fault detection method of air source heat pump system using convolutional neural network for energy saving," *Energy*, vol. 187, p. 115877, 2019.

[13] J. Chen, "Improved maximum likelihood location estimation accuracy in wireless sensor networks using the cross-entropy method," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 1325–1328.

[14] J. Yang and G. Yang, "Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer," *Algorithms*, vol. 11, no. 3, 2018.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Robotics and Machine Learning*, p. 533, 2014.

Paper B

Robustness analysis of PCA-SVM model used for fault detection in supermarket refrigeration systems

Zahra Soltani, Kresten Kæjr Sørensen, John Leth, Jan Dimon Bendtsen

The paper has been published in the
2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pp. 1–6, 2021.

The layout has been revised.

Abstract

Supermarket refrigeration systems represent an important type of energy demanding appliances, which is in such widespread use that any development in the associated technology can have a huge impact on general health and global warming. Using automatic fault detection and diagnosis may for instance improve energy efficiency and reduce food waste as well as reduce expenses for the supermarket owners. In this paper, three model-free classification algorithms are tested on faulty/non-faulty data obtained from an actual refrigeration system. It is found that support vector machines (SVM) are able to classify fan faults in a real refrigeration system with near-100% classification accuracy, independent of the number of input variables. The classification performance and robustness against an unseen operation mode, low-resolution data, noisy data, and data of different operating points is tested for three different classifier configurations. The results show Principle Component Analysis (PCA)-SVM is highly robust to different operating points, disturbances, and gives the best computational efficiency, as it is able to reduce the feature space to only two dimensions. It is concluded that while all of the examined methods are insensitive to noise, and effective in terms of detecting faults from relatively small amounts of data, overall, PCA-SVM is slightly more computationally efficient.

1 Introduction

Recently, data acquisition and data monitoring have become a part of business competitiveness in many industries, and the availability of data enables manufacturers to have more efficient and reliable systems. Automatic fault detection and diagnosis is one of the ways that ensure more efficient and reliable systems. In refrigeration systems (RS), for food storage, it is crucial to stay within a narrow temperature band; and therefore, it is important to detect faults before they turn into a system breakdown. If the airflow over the evaporator is reduced due to a faulty fan, it will normally not be noticed until the room temperature cannot be kept at its set-point. Traditional fault detection in supermarket refrigeration systems requires many expensive sensors and provide only limited identification of the root cause. Therefore, data-driven Automatic Fault Detection and Diagnosis (AFDD) of such systems is desired. One of the main challenges in designing automatic fault detection systems is that RS controllers, like the ones made by Bitzer Electronics, are used on many different refrigeration systems that exhibit different dynamical behaviour.

Different algorithms have been applied for fault detection and diagnosis (FDD) of refrigeration systems such as [1–5]. In [3], the rule-based fault classifier achieved higher effectiveness than data-driven models when the FDD performance index is a controlled variable. The components characteristics and operations anomalies for different types

of SRS are studied in [2]. In this study, three sources of the industry including expert surveys, advisory messages such as alarms, and service calls are considered. This information can be used for expanded development of fault detection models. A Convolutional Neural Network was applied for fault detection of refrigeration systems in [4]. This algorithm achieved more than 99 % accuracy in fault classification. The results show that the model can be trained better using low-resolution data. However, as CNN is a deep learning model, it requires high amount of data and computation capacity. A Gaussian mixture model is used in [5], and data dimensions are reduced using Principal Component Analysis (PCA). This model classified four types of faults in Air-conditioning systems with about 99% accuracy, and the running time is reduced more than ten times. One of the binary classifiers that can classify the data based on a low number of samples is SVM. Compared to many types of ANN algorithms SVM has both fast computation and good accuracy. SVM is used in many fields for data classification see [6–8], condition forecasting [9], and fault detection [10]. SVM is also used in [11] for fault detection in vapor compression refrigeration systems.

In this work, SVM is used to distinguish between a system with evaporator fan fault and a functional system in different operation points while data from evaporation side is not available. It is infeasible to design a bespoke fault detection algorithm for use in every supermarket or to have the technician set it up correctly for each new system and therefore, an automated adaptive fault detection method is required. The topology of the refrigeration systems controlled by the Bitzer condensing unit is generally the same, but they may vary in size and operational set-point. In other words, the challenge addressed in this paper is to design a fault detection algorithm that works effectively for 'generic' cooling systems, where the availability of particular combinations of signals cannot be guaranteed. In this paper, we present a method that is robust against the aforementioned types of variations. We show that, through careful selection of the inputs to the classifier, the amount of computation required can be reduced and that PCA can be used as a form of normalization of faulty data acquired at different set-points.

The remainder of this paper is structured as follows. Section 2 introduces SRS background and fault detection methodology used in this study. SVM classifier and PCA are explained in section 3. The models structure and training sensitivity are studied in Section 4. Afterwards, robustness analysis, and comparison of the classifiers are introduced in section 5. In Section 10, the results of the work are concluded.

2 Supermarket refrigeration systems

Supermarket refrigeration systems normally use the vapor-compression refrigeration cycle in which heat is moved from a low temperature space to higher temperature ambient air. The heat transferring in the cycle leads to phase change from liquid to vapor and vice versa. Fig. B.1 represents an example of SRS which is later used in this paper. Nomenclature related to the figure is described in Table B.1. SRS might require several

controllers for evaporation units and condensing units depending on the supermarket requirements and conditions. In Fig. B.1, the system has two controllers that control condensing and evaporation side separately. In SRS, an evaporation unit controller ($Ctrl_{evap}$) controls superheat temperature (T_{sh}) or suction temperature (T_{suc}) to regulate the evaporator performance. A condensing unit controller ($Ctrl_{cond}$) controls cold room temperature by adjusting compressor work.

The evaporator fan plays a key role in transferring heat from the goods to the evaporator surface and consequently the refrigerant. Moreover, it circulates the air in the cold room to ensure an even temperature. An evaporator fan anomaly leads to uneven room temperature, wrong temperature readings by the sensors, higher power consumption, and finally food spoilage. Therefor, early fault detection algorithm is demanded to prevent those consequences. For an AFDD algorithm, data from a condensing unit is required.

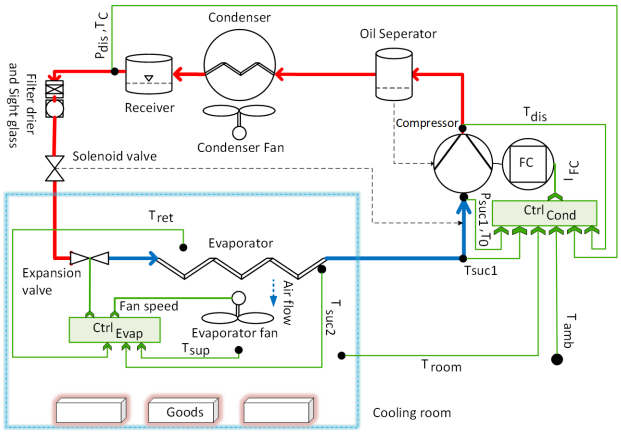

Fig. B.1: Schematic of the laboratories' SRS [4].

In this study, data of the normal condition is called *non-faulty*, and data when the fan is defective is called *faulty* data. Variations in the data are necessary to ensure that the developed fault detection model is robust against variations in refrigeration system dynamics. Such variations include evaporators size, air temperature set-points and suction super-heat. In this work, cooling load varied from 6 to 17 kW, the set-point is changed from 1 to 12 °C, and as a consequence, the compressor speed varied from 33 to 80 Hz. As seen in Fig B.1, the laboratory set up has two evaporator fans. Fan fault is emulated in the laboratory set up so as one out of two evaporator fans is defective. In all data sets 14 measurements are logged from $Ctrl_{cond}$ which are relevant to show the system characteristics. The collected data is fed into the AFDD algorithm, which is described below.

Table B.1: Symbols used in the Fig. B.1 [4].

Symbols	description	SI unit
T_{room}	cooling room temperature (sensor)	$[°C]$
T_{amb}	ambient temperature (sensor)	$[°C]$
$T_{suc1,2}$	suction temperature (sensor)	$[°C]$
T_0	saturation evaporation (sensor)	$[°C]$
P_{suc}	suction pressure (sensor)	$[Pa]$
T_{dis}	discharge temperature (sensor)	$[°C]$
$Pdis$	discharge pressure (sensor)	$[Pa]$
T_c	saturated condensing temperature	$[°C]$
T_{ret}	returned air temperature (Sensor)	$[°C]$
T_{sup}	supplied air temperature (Sensor)	$[°C]$
I_{FC}	converter current	$[A]$
FC	frequency converter	[-]
$K_{\dot{m}}$	proportional mass flow rate	$[kg/m^3s]$
$Ctrl_{Evap}$	evaporator controller	[-]
$Ctrl_{Cond}$	condenser controller	[-]

2.1 Fault detection methodology

In this paper, three different fault classifiers are presented: SVM classifier using all available signals of relevance to the system characteristics, SVM classification using signals selected by experts based on system knowledge, and PCA-SVM classification in which PCA is used for feature extraction. Making a single fault detection algorithm that is capable of handling system variations requires that the features that are most important for detecting the fault are extracted from the data and normalized before being passed to the classifier. In this paper, the feature extraction or signal selection are tested both manually and automatically.

As for the second methodology, the most relevant signals are selected manually by experts. Reducing the number of inputs can be effective both for recording the data and also for classification computation. However, it is of course detrimental if any information-rich signals are removed. In this work, the most relevant signals that represent the system characteristics during fan fault detection are: P_{suc}, T_{sh}, compressor speed (V_{cpr}), and proportional mass flow rate ($K_{\dot{m}}$). Even though this methodology is computationally more efficient than the one that used all signals, it is vastly dependent on experts knowledge.

The last methodology is feeding all available signals to the PCA algorithm which extracts the most important information of the data and reduces its dimensions. Afterward, the reduced-order data is classified by SVM. This methodology should enable

a simpler classifier to distinguish between faulty and non-faulty data from a range of systems with varying characteristics.

3 Methods

3.1 SVM classifier

Support Vector Machines is a type of supervised learning method used for classification purposes. Let a set of data be given as $\mathcal{D} = \{(x_i, y_i)|x_i \in \mathbb{R}^k, y_i \in \{-1, 1\}\}_{i=1}^n$ where k is the dimension of the samples x_i, y_i is the corresponding class and n is the number of the samples. If $\mathcal{D}$ is linearly separable, it is possible to select two parallel hyperplanes that separate the two classes of data, such that the distance between them is as large as possible. The region bounded by these two hyperplanes is called the *margin*, and the maximum-margin hyperplane is the hyperplane that lies halfway between them, as illustrated in Fig B.2. A hyperplane h is the set of points $x \in \mathbb{R}^k$ satisfying an equation of the form

$$w \cdot x + b = 0 \tag{B.1}$$

where $\cdot$ is the standard vector dot product, $w \in \mathbb{R}^k$ (a.k.a. *weights*) is orthogonal to h, and $b \in \mathbb{R}^k$ is an offset (a.k.a. *bias*). Notice that the two classes are separated by two parallel hyperplanes h_1 and h_2 defined by

$$\begin{aligned} h_1 &: w \cdot x + b = 1, \\ h_2 &: w \cdot x + b = -1. \end{aligned}$$

Since h_1 and h_2 are parallel, they share the same w, and the distance between them is $2/\|w\|$. The distance between h_1 and h_2 is thus maximized by solving the following constrained optimization problem [12]:

$$\begin{aligned} &\min\ f(w) = \|w\| \\ &\text{s.t.} \\ &\begin{cases} w \cdot x_i + b \geq 1 & \text{if } y_i = 1 \\ w \cdot x_i + b \leq -1 & \text{if } y_i = -1 \end{cases} \end{aligned} \tag{B.2}$$

In SVM, the hyperplanes are typically represented via *perceptrons* parameterized by the weight and bias vectors w as

$$H(x_i) = \text{sign}(w \cdot x_i + b). \tag{B.3}$$

The parameters are adjusted using an update rule in order to find the correct classification for each sample. Here, the technique of Lagrange multipliers is used, allowing the

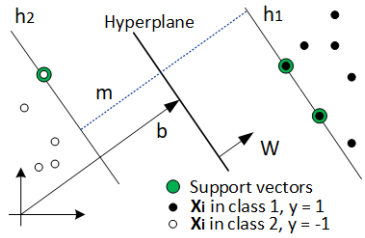

Fig. B.2: Maximum margin (m) and optimal hyperplane.

minimization problem (B.2) to be rewritten as:

$$\begin{aligned} &\min W(\alpha) = \Sigma_{i=1}^{n}\,\alpha_i - \frac{1}{2}\sum_{i}^{n}\sum_{j}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \\ &\text{s.t.} \\ &0 \leq \alpha_i \leq C \\ &\sum \alpha_i y_i = 0 \end{aligned} \tag{B.4}$$

where $\alpha_i \geq 0, i = 1, \ldots, n$, are Lagrange multipliers, the constant C is a bound on the multipliers determining how SVM deals with classification errors, and $K : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ is a so-called *Kernel function*. The parameter C is a trade-off between narrow and wide margins. Selecting a too small value for C (corresponding to a wide margin) might result in hyperplanes that can not classify a validation data set, whereas hyperplanes resulting from too large C (narrow margin) might not handle noisy outliers well.

In some cases, the data set might not be linearly separable in its original representation. Then, it is often possible to transform the data by a kernel function such that the classes become separable in the transformed representation. Several types of kernel functions, such as polynomials, Radial Basis Functions (RBF), etc. may be used. A popular choice is the RBF kernel function, described by

$$K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2} \tag{B.5}$$

where the bandwidth parameter γ is inverse of the variance of standardized samples which scales the distance between two samples. There are a number of heuristics to determine C and γ as hyperparameters of the optimization, which can be found in [12].

3.2 PCA

It is often the case in practice that some of the features are correlated with the others, thus providing less useful information for the classification. Principal Component Analysis (PCA) is a method that analyzes high dimensional data and identifies correlations

among the data entries (features). PCA then projects the data down to a lower dimensional representation in which important relations between features, and other relevant information of the data set are preserved, but, unimportant information is discarded. The basis of this new representation, called *principal components*, is orthogonal by construction, as it is the span of eigenvectors of the covariance matrix of $\mathcal{D}$. The main advantage of PCA in this particular application is that it removes correlated features that do not make any contribution to the classification.

Correlation among the parameters can be identified by computing the covariance matrix $R_{xx}(\mathcal{D}) \in \mathbb{R}^k \times \mathbb{R}^k$. From $R_{xx}(\mathcal{D})$, we compute the k eigenvectors ν needed for projecting the k-dimensional samples onto the subspace spanned by the principal components. The eigenvectors are sorted by descending eigenvalues, and only the eigenvectors corresponding to the largest $m < k$ eigenvalues are used for the projection. Finally, a new data set $\mathcal{D}_{\text{pca}}$ can be obtained from the original data by computing

$$\tilde{x}_i = V(V^T V)^{-1} V^T x_{s_i}, \quad i = 1, \ldots, n \tag{B.6}$$

with $V = [\nu_1 \ \cdots \ \nu_m] \in \mathbb{R}^{k \times m}$, yielding $\tilde{x}_i \in \text{span}\{\nu_1, \ldots, \nu_m\}$ for all $i = 1, \ldots, n$.

4 Model training

As outlined above, first, an SVM classifier is designed for 14D and 4D input data. Then, the PCA-SVM algorithm is presented and compared with the two other proposed classifiers. The sensitivity of the classifiers during training phase is investigated against different sample rates and different data lengths (number of samples).

4.1 Training and Validation

In this work, data is classified into two categories of faulty and non-faulty. Fourteen measurements are logged from the condensing unit controller and fed into SVM when all information of the system is used. In this data set, neither human nor an algorithm selects the relevant signals. The SVM classifier used RBF kernel function with optimized hyperparameters of $C = 10$ and $\gamma = 1$. The result of SVM using 14D data represents 98% accuracy for training data and 100% accuracy for validation data classification. Afterwards, four of the aforementioned measurements are selected and supplied into a SVM algorithm. 98% classification accuracy in the training phase, and 100% classification accuracy in the validation phase is obtained. In this case, RBF kernel function is used with optimized hyperparameters, $C = 100$ and $\gamma = 1$.

As for the third algorithm, PCA is used to obtain the most correlated features. The scree plot in Fig. B.3 illustrates the variation that each principal component accounts for in percentage. Therefore, the first two principal components, which has the most variance, is selected. Fig. B.4, represents the training data classification using PCA-SVM classifier. The contour maps shows the choice of the decision boundary between

the two classes of data using Radial basis kernel function (RBF), $\gamma = 1$, and $C = 100$. Fig. B.4, represents the training data classification using PCA-SVM classifier. The

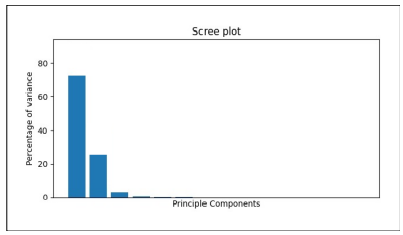

Fig. B.3: The percentage of variation that each Principal component accounts for

contour maps shows the choice of the decision boundary between the two classes of data using RBF kernel function, $\gamma = 1$, and $C = 100$. The faulty data is bounded

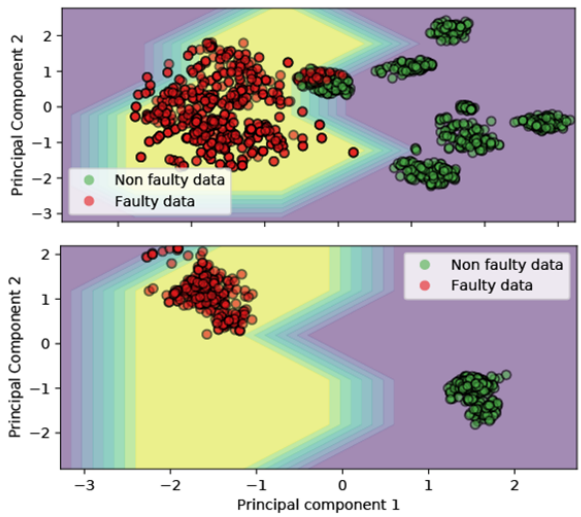

Fig. B.4: The top plot, training result of PCA-SVM model. The bottom plot, the validation result of PCA-SVM.

by yellow surface and the boundary become loser until a surface that belongs to the non-faulty data indicated by purple. Here, the data could have been classified with looser boundaries or more hyperplanes. However, the more restricted margin is selected due to two reasons. First, as seen in Fig. B.4, in the top, the non-faulty data is more varied and distributed differently. Therefore, misclassification of unseen non-faulty data is avoided by more restriction for faulty data. Secondly, the smallest false positive

rate is ensured, which is desired in the industry. In the bottom plot of Fig. B.4, the validation data set is transformed into the principal components of the training data, which causes the different positions of the validation data compared to the training data. The classifier detects the training non-faulty data with 98% accuracy and faulty data with 97% accuracy. The validation data is classified with 100% accuracy for both faulty and non-faulty data. The validation result was more accurate than the training due to the distribution and overlap of the training data, which are less prominent in the validation data. The distribution of the training data is due to using data of different operation conditions while the validation set is taken only from one operating condition.

4.2 Training data sensitivity

In SVM, the number of samples to be used depends on the number of input measurements, meaning that if higher dimensional data is selected, the data set should be increased as well to achieve better performance. However, as the computation efficiency is important in this work, further tests with longer data set is ignored. Instead, 4D data with different lengths are tested for the SVM training.

In each test, training data with different sample rates are proposed. Training data is down-sampled from 1 Hz to 0.3, 0.1, 0.03, and 0.01 Hz. Here, it is not possible to analyze lower sample rates than 0.01 Hz, due to the limited data length. Table B.2 represents the accuracy of the SVM classification to the various training data. In this analysis a specific validation data set is used which has different operating condition than training data. Remark, in this table, the length of the data is the number of samples of each class.

Table B.2: Training data length and resolution analysis

Length	Sample rate [Hz]	Training time (s)	Accuracy [%]
1800	1	0.57	93
	0.1	0.65	93
	0.01	0.63	93
900	1	0.09	99
	0.1	0.09	99
	0.01	0.1	99
300	1	0.07	94
	0.1	0.08	94
	0.01	0.07	94

From Table B.2, it can be recognized that different sample rates do not have effect neither on accuracy nor running time whereas data length has a considerable effect on both the accuracy and running time. It is found that the best training data length is

about 900 samples for each class of data. The number of samples need to be sufficient enough to cover all information of the data. Therefore, an insufficient number of data leads to misclassification. Moreover, the classifier needs to handle too many outliers if it receives too large number of samples. In addition, by doubling the number of samples the training time increased about 60%.

5 Robustness analysis

As mentioned in Section I, it is necessary to have an algorithm that is robust to different system configurations and operating conditions. Therefore, robustness tests for SVM and PCA-SVM classifiers are done. Note that in this section, data length is 900 samples and sample rate is 1 Hz for the training set.

5.1 Validation data resolution

In this test the training and validation data have different sample rates. Validation data with 1 Hz sample rate for all three classifiers obtained the same accuracy about 100%. Then, the validation data is down-sampled from 1 Hz to 0.1 and 0.01 Hz. However, the classification accuracy remains the same as using original data. The results shows that SVM is a robust classifier against data resolution as the same results are obtained after down-sampling of the original validation data. This test illustrates the validation data is accurately classifiable independent of the data resolution. A classifier trained in a specific sample rate can be used to classify the fault in a variety of RS with different sample rates.

5.2 System variations

To investigate robustness towards RS variations, the validation data was changed by adding noise, static perturbations (offset), and an operational disturbance as seen as On/Off operation of the compressor in RS. Every type of test is done 20 times to ensure the results. Table B.3, illustrates the classification results of system variations tests. The changes to the data was exacerbated compared to data from the field to ensure that the classifiers can handle a wide range of refrigeration systems. The noise is random with normal distribution $\mathcal{N}(0, 2)$ and values ranged [-4, 4] °C. As shown on Fig. B.5, when adding noise to the data, non-faulty and faulty data overlap in some of the measurements and become harder to separate. Different system configurations and operating conditions in SRS can be considered as perturbations of the data assuming all or some of correlations between the measurements are preserved. On Fig. B.6, the classifier's results for perturbed data is shown. Here, random offset of the superheat temperature in the range [-5 to -2]$°C$, and [2 to 12]$°C$ is applied. In fact, perturbation might not make a huge impact on the result when using PCA as long as correlation

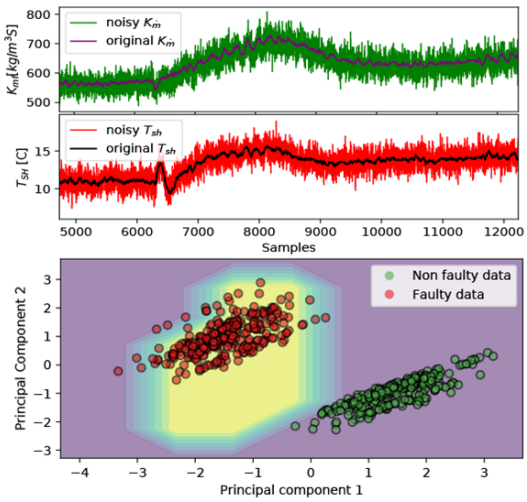

Fig. B.5: The top plot , noisy data example (fault occurs at sample no. 6300); at the bottom, Noisy data classification using PCA-SVM.

of the data does not change. Comparing the classifiers results in Table B.3, the PCA-SVM classifier is more robust against perturbed data or different operation conditions. In SRS, when the temperature of the goods are on set-point, low cooling capacity is required to keep the goods at the same temperature. Thus, the SRS operation mode may alternate between stopped and running modes. A slow and periodic disturbance has been added to the data to simulate On/Off mode of operation for the compressor seen on Fig. B.7. Table B.3 represents better classification using PCA-SVM than two other classifiers in the on/ off mode.

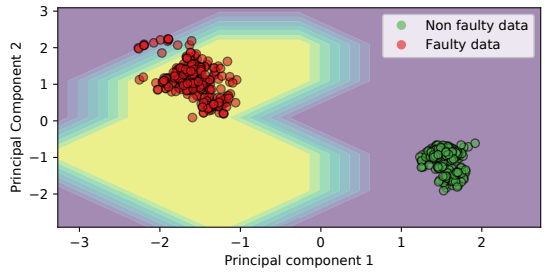

Fig. B.6: Validation of perturbation test using PCA-SVM classifier.

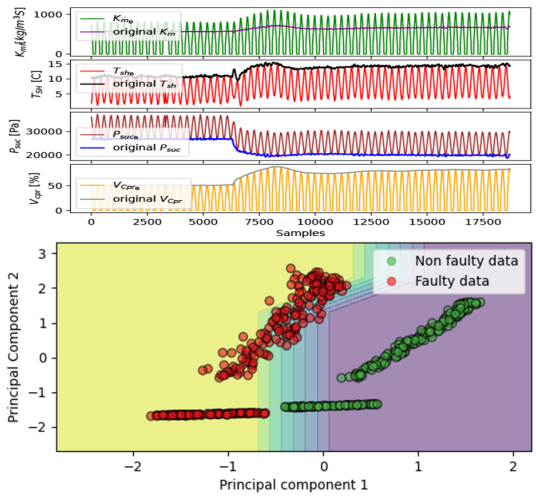

Fig. B.7: The top plot, an example of data when disturbance is added; the bottom plot, validation of disturbance test using PCA-SVM.

6 Conclusion

In this study, it was shown that a SVM classifier can identify a fault in evaporation side using data from condensing unit with high accuracy, in both training and validation process, independent of the data resolution. It was shown that it is possible to do fault detection on refrigeration systems using Machine learning with lower amount of expert effort which is expensive and time consuming. Three models are proposed to classify the data using SVM classifiers. The difference among these classifiers are their inputs which were raw data from the controller for the first (14D) model, the most relevant measurements for the second (4D) model, and PCA transformed data for the third model. The classifiers are highly robust to different data sample rates as long as the dynamics of the system is preserved. PCA-SVM can overcome the significant difficulties that unseen data introduces for the classifiers such as noise, perturbation, disturbance and different running modes. PCA-SVM is more robust against system variations and about 25% more computationally efficient than SVM without dimension reduction.

Another advantage of the PCA-SVM algorithm is that it can be separated into two parts; a PCA algorithm, and an SVM algorithm. PCA can be processed in the controller hardware, and the transformed data with low dimensions can be sent to the third party for the fault classification. Therefore, PCA-SVM can be considered as the most accurate and cost-effective classifier among those three proposed classifiers.

Table B.3: The classifiers robustness tests.

	Algorithm	Non faulty[%]	Faulty[%]	Run time[s]
Noisy	14D SVM	98.5 -99.6	98 -99.4	0.31
	4D SVM	98 -100	98 -99.4	0.24
	PCA-SVM	98 -100	98 -99.6	0.25
Perturbed	14D SVM	89-100	97-100	0.32
	4D SVM	99.2-100	99-100	0.24
	PCA-SVM	100	100	0.23
On/Off	14D SVM	50-60	53-60.5	0.33
	4D SVM	55-60	54-61	0.25
	PCA-SVM	85-86	95.5-96.4	0.25

References

[1] Y. Guo and H. Chen, "Fault diagnosis of vrf air-conditioning system based on improved gaussian mixture model with pca approach," *International journal of refrigeration*, vol. 118, pp. 1–11, 2020.

[2] A. Behfar, D. Yuill, and Y. Yu, "Supermarket system characteristics and operating faults (rp-1615)." *Science & Technology for the Built Environment*, vol. 24, no. 10, pp. 1104 – 1113, 2018.

[3] ——, "Automated fault detection and diagnosis for supermarkets–method selection, replication, and applicability," *Energy and Buildings*, vol. 198, pp. 520–527, 2019.

[4] Z. Soltani, K. K. Soerensen, J. Leth, and J. D. Bendtsen, "Fault detection of supermarket refrigeration systems using convolutional neural network," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 231–238.

[5] Y. Guo and H. Chen, "Fault diagnosis of vrf air-conditioning system based on improved gaussian mixture model with pca approach," *International Journal of Refrigeration*, vol. 118, pp. 1 – 11, 2020.

[6] C. Zhou, J. G. Chase, and G. W. Rodgers, "Support vector machines for automated modelling of nonlinear structures using health monitoring results," *Mechanical systems and signal processing*, vol. 149, pp. 107 201–, 2021.

[7] "Modeling and optimization of a light-duty diesel engine at high altitude with a support vector machine and a genetic algorithm," *Fuel (Guildford)*, vol. 285, pp. 119 137–, 2021.

[8] A. Skariah, P. R, R. R, and B. C. R, "Health monitoring of rolling element bearings using improved wavelet cross spectrum technique and support vector machines," *Tribology International*, vol. 154, p. 106650, 2021.

[9] S. Rafiee-Taghanaki, M. Arabloo, A. Chamkalani, M. Amani, M. H. Zargari, and M. R. Adelzadeh, "Implementation of svm framework to estimate pvt properties of reservoir oil," *Fluid Phase Equilibria*, vol. 346, pp. 25 – 32, 2013.

[10] J. Liu, Y.-F. Li, and E. Zio, "A svm framework for fault detection of the braking system in a high speed train," *Mechanical Systems and Signal Processing*, vol. 87, pp. 401 – 409, 2017.

[11] 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, pp. 295–313, 2010.

[12] M. Jordan, J. Kleinberg, and B. Schölkopf, *Support Vector Machines*, 1st ed., ser. Information Science and Statistics. New York, NY: Springer-Verlag, 2008.

Paper C

Fault detection and diagnosis in refrigeration systems using machine learning algorithms

Zahra Soltani, Kresten Kæjr Sørensen, John Leth, Jan Dimon Bendtsen

The paper has been published in the
International journal of refrigeration, vol. 144, pages 34–45, 2022 *Elsevier*,

Appendix A is added to this assertion which is notified to the paper's publisher, but it is not applied to this paper at the moment of submitting the thesis. This appendix includes the important corrections of this paper.
The layout has been revised.

Abstract

The functionality of industrial refrigeration systems is important for environment-friendly companies and organizations, since faulty systems can impact human health by lowering food quality, cause pollution, and even lead to increased global warming. Therefore, in this industry, there is a high demand among manufacturers for early and automatic fault diagnosis. In this paper, different machine learning classifiers are tested to find the best solution for diagnosing twenty faults possibly encountered in such systems. All sensor faults and some relevant component faults are simulated in a high fidelity Matlab/Simscape model of the system, which has previously been used for controller development and verification. In this work, Convolutional Neural Networks, Support Vector Machines (SVM), Principal Components Analysis-SVM, Linear Discriminant Analysis-SVM, and Linear Discriminant Analysis classifiers are compared. The results indicate that the fault detection reliability of the algorithms highly depends on how well the training data covers the operation regime. Furthermore, it is found that a well-trained SVM can simultaneously classify twenty types of fault with 95% accuracy when the verification data is taken from different system configurations.

1 Introduction

Machine Learning (ML) is a common term for many processing methods used for data-driven tasks. The main intention of ML is to enable computers to learn, predict, or decide on an unseen data without human assistance [1]. In the 2010s, rapid development of processors, IoT, and an increasing amount of generated data paved the way for large improvements in ML capabilities. Thus, the popularity of ML increased exponentially in many industries. Machine learning is used in various contexts, such as computer vision, text classification, fault detection, language processing, image recognition, and so forth.

The idea of using ML for fault detection and diagnosis dates back to the 1980s where the existing ML methods were not as efficient as specialized experts. However, the technologies have been improved, and as of today, the availability of powerful programming tools and algorithms for self-learning allow computers to make strategic decisions and even diagnose new events [2].

In particular, ML-based methods have been studied for fault detection and diagnosis (FDD) in different fields with promising results. For instance, ML is used for fault detection in brushless synchronous generators in [3], in water distribution network [4], in age intelligence systems [5], and in high-temperature super conducting DC power cables [6]. In [7], several supervised ML algorithms are compared for FDD in photovoltaic

systems. In [7], data from non-faulty condition and five different faulty conditions are used both for training and test; and the results confirm that supervised learning algorithms can be used for fault detection and ease the FDD procedure. Moreover, machine learning models are compared for sensor fault detection in [8], in which five types of sensor faults are emulated, namely, drift, bias, precision degradation, spike, and stuck faults.

For fault detection in office building systems, various data mining methods, in particular, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Kernelized Discriminant Analysis (KDA), semi-supervised LDA, and semi-supervised KDA have been compared in [9]. In [10], different component faults in a rotating machine are classified using a Convolutional Neural Network (CNN) algorithm. According to [11], in many industrial applications, good system models are difficult or even impossible to obtain due to the system's complexity or large numbers of configurations involved in the production process. The refrigeration industry is not an exception, as the system configuration varies based on different owners' demands. Hence, model based FDD is often sensitive to model parameters in such a way that small changes in the system may lead to a poor fault detection response. In such cases, ML can be a viable approach to handling unseen situations when well trained.

In [12], a CNN model is used for evaporator fan fault detection in supermarket refrigeration systems. The same system configuration and information are used in [13] to classify the same fault and investigate the robustness of the fault detection model. However, instead of CNN, shallow learning Support Vector Machines SVM and PCA-SVM classifiers are used. In [14], SVM and PCA-SVM are studied for the detection of 8 type of faults in a simulated vapour-compression refrigeration system in which PCA-SVM achieved a better result compared to SVM and back-propagation neural network.

In the refrigeration industry, good performance of a fault detection algorithm can be defined as high classification accuracy, low computation time, and low false positive rate. High classification accuracy ensures an accurate fault description for the technicians for quick troubleshooting, while low computation time is important because it lowers the detection time and the hardware cost. A low false positive rate increases the reliability of the fault detection model and results in lower expenses regarding service call rate. Therefore, it is essential to evaluate the FDD algorithms based on these factors.

Because of increasing usage of digitalization in refrigeration systems (RS), many companies aim for improving existing FDD performance by utilising various data. As mentioned above, FDD algorithms perform satisfactorily in many other applications; thus, data driven FDD algorithms are selected and evaluated in this work. That is, we evaluate and optimize various FDD algorithms for the purpose of selecting the best classifier for use in RS industry applications.

The main contributions of this study is summarized below:

- A deep learning and several shallow learning classifiers are proposed for detecting and diagnosing twenty types of faults in RS.

- Importance of training data qualification regarding data variation and features selection is illustrated.
- All of the proposed classifiers are compared regarding classification accuracy, computation time and false positive rate.
- The best approach from an industrial perspective is proposed to detect a faulty system and localize the fault.

In this study, all sensor faults and some component faults are simulated using a high fidelity RS model. The model is already in use at Bitzer Electronics to develop and verify control algorithms. Notice that we will restrict our attention to steady state operating conditions, which are commonly encountered in industrial application such as reefer containers, cold storage houses and so on. It is acknowledged that transient operation is important in many applications as well, e.g., in supermarket refrigeration systems. However, transient behavior presents its own set of unique challenges, and is considered out of scope of this work.

The faults include positive and negative offsets in sensors as well as specific component faults; the faults are detailed in section 2. Three classifiers, namely CNN, SVM, and LDA, are compared to diagnose every selected fault. For pre-processing of the input data LDA and PCA are compared.

The results indicate that the SVM classifier is the superior method, being able to diagnose all classes with 100% classification accuracy except non-faulty and malfunctioning of expansion valve conditions which are diagnosed with 98% and 96% classification accuracy, respectively. The LDA and LDA-SVM classifiers are capable of detecting the faulty condition with 100% classification accuracy. However, these models have poor performance regarding robustness as a significant drop in classification accuracy is observed. Finally, CNN and PCA-SVM show a general lack in performance.

The remainder of this paper are structured as follows. First, refrigeration systems background and specification, as well as data acquisition and its specification, are introduced in section 2. Then, in section 3, the mathematical approaches of the classifiers mentioned above are explained. Afterwards, the specification of each model and the result of the classification is presented in section 4. Finally, the work is concluded in section 10.

2 Background

In general, RS are used to cool down the goods inside of an insulated room, which is called a cold room, by transferring the heat to the environment. Fig. C.1 illustrates a RS in which the refrigerant runs through the pipes. In each refrigeration cycle, heat is absorbed and dissipated. The compressor receives low pressure, low temperature refrigerant gas and releases high pressure, high temperature gas to the inlet of the

condenser. The condenser is responsible for dissipating the refrigerant heat to the ambient environment, and finally gives out liquid refrigerant at high pressure while the temperature decreases. Afterwards, an expansion valve decreases the pressure of the refrigerant. Low pressure, low temperature refrigerant enters the evaporator pipes in order to absorb the heat from the cold room environment. Thus, the refrigerant changes phase from liquid to gas before reaching the compressor.

Defective components or sensors in RS lead to high power consumption, air pollution, wear and tear of the components, and/or food waste. RS have the best efficiency when everything is nominal. Thus, when faults occur, the system might deviate from the peak efficiency point. By some of the fault, the system runs outside of its permitted envelope, some of the faults lead to wear and tear of the components due to high temperature, too little lubrication, and too high pressure on the components. Late fault detection may cause the temperature of the refrigerated goods to exceed the permitted limits. Therefore, early fault detection in RS ensures maintaining the required quality of refrigerated goods such as food products or medicine, and preventing excessive maintenance and spoilage cost.

The high fidelity model used by Bitzer Electronics is presented in Fig. C.2. In this model, a two-stage semi-hermetic reciprocating compressor is simulated with operating speed in the range 25-87 Hz. Here, compressor cooling capacity (V_{cpr}) is defined as compressor operating speed in percentage. Therefore, compressor speed under 25 Hz and full speed operation of 87 Hz are defined as 0% and 100% compressor cooling capacity, respectively. The refrigerant type is R134a, and an electrical expansion valve is simulated. Maximum cooling capacity of the cold room is 17 kW at 10 °C ambient temperature (T_{amb}) and 5 °C cold room temperature (T_{room}). The controller is designed so as it controls over opening degree of expansion valve ($vexp$) using superheat temperature (T_{sh}) measurements as an input. T_{sh} is the difference between the refrigerant evaporation temperature (T_0) and suction gas temperature (T_{suc}). In addition, V_{cpr}, evaporator fan speed (V_{evap}), condenser fan speed (V_{cond}), are controlled using the mentioned controller inputs in Fig. C.2. In this paper the supply temperature (T_{sup}) is the same as cold room temperature (T_{room}) and used as set point in the simulation model. Thus, set point is the temperature of the air after transferring heat to the refrigerant.

In Fig. C.2, the main components of the model are presented with grey blocks. The red blocks indicate some of the fault inputs which are added to the corresponding parameters. Twenty types of faults are simulated, including positive and negative offsets in sensors as well as a number of component faults; the faults are described in Table C.1. When collecting a data set, the model is first run with no effect of the red blocks, thus producing non-faulty data. After logging sufficient non-faulty samples, one fault is applied to the model and data collection continues. Simulation of some of the faults such as pressure sensors offset and T_{dis} sensor offset, are not visible in Fig. C.2, since they are simulated inside of the relevant block diagrams.

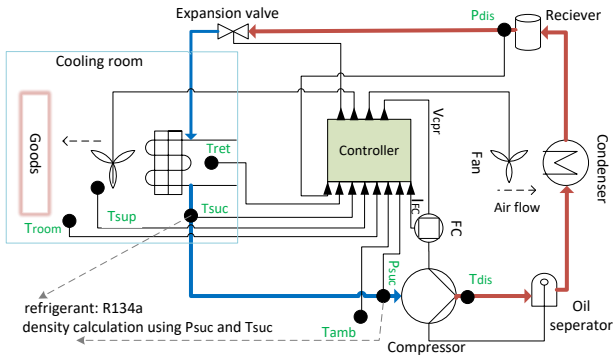

Fig. C.1: Schematic of a refrigeration system

2.1 Data acquisition

Machine learning models learn based on input information. Thus, the quality of training data is an essential factor. The training data should contain sufficient information to have a generic algorithm to make a correct decision when receiving a new observation. Using simulated data for training phase can, in fact, improve the verification result since it firstly allows data collection in different operating conditions, and secondly, data of specific faults can be correctly labeled, and finally, we ensure that the training data is not taken from an already faulty system with unwanted or unknown fault.

To prevent overfitting the model, the input data needs to be taken from various operating conditions in an acceptable range and under the same operation conditions for each fault. That is, the model has to be able to deal with operational variations. Generally, in RS, operations vary based on several factors, such as required temperature set point, compressor cooling capacity or heat load, compressor type, ambient temperature, etc. In this work, various data sets from different operating conditions are taken as training data. As shown in Fig. C.3, the set point is changed in the range 0 to 15 °C, and the heat load in the cooling room varies in the range 3 to 20 kW to obtain compressor speeds variation. Another data set is taken in which, besides set point and heat load, the T_{amb} is varied; therefore, the data is referred to as having large operation condition range. In this data set, T_{amb} is varied in the range 10 to 30 °C to investigate how the classification accuracy differs if training data includes more variations. Then, the verification data set is collected using different operating conditions from the training conditions to investigate how the model performs classification in an unseen operation condition, see the blue block in Fig. C.3.

Each fault in the system is considered a class. As introduced in Table C.1, twenty faults are taken into account in this work which are all observed in the real systems.

Table C.1: fault types and descriptions

Label	Component	Fault limits
1	T_{suc} sensors positive offset	$2°C$ more than expected value
2	T_{sup} sensors positive offset	$2°C$ more than expected value
3	T_{ret} sensors positive offset	$2°C$ more than expected value
4	T_{dis} sensors positive offset	$2°C$ more than expected value
5	P_{dis} sensor positive offset	100000 Pa more than expected value
6	P_{suc} sensor positive offset	20000 Pa more than expected value
7	compressor low performance	less than 80% of the expected mass flow
8	loose expansion valve	more than 120% of the commanded opening degree
9	evaporator fan low performance	less than 80% of the commanded fan speed
10	condenser fan low performance	less than 80% of the commanded fan speed
11	T_{suc} sensors negative offset	$2°C$ less than expected value
12	T_{sup} sensors negative offset	$2°C$ less than expected value
13	T_{ret} sensors negative offset	$2°C$ less than expected value
14	T_{dis} sensors negative offset	$2°C$ less than expected value
15	P_{dis} sensor negative offset	100000 Pa less than expected value
16	P_{suc} sensor negative offset	20000 Pa less than expected value
17	broken compressor	less than 20% of the expected mass flow
18	blocked expansion valve	less than 80% of the commanded opening degree
19	broken evaporator fan	less than 20% of the commanded fan speed
20	blocked condenser fan	less than 20% of the commanded fan speed

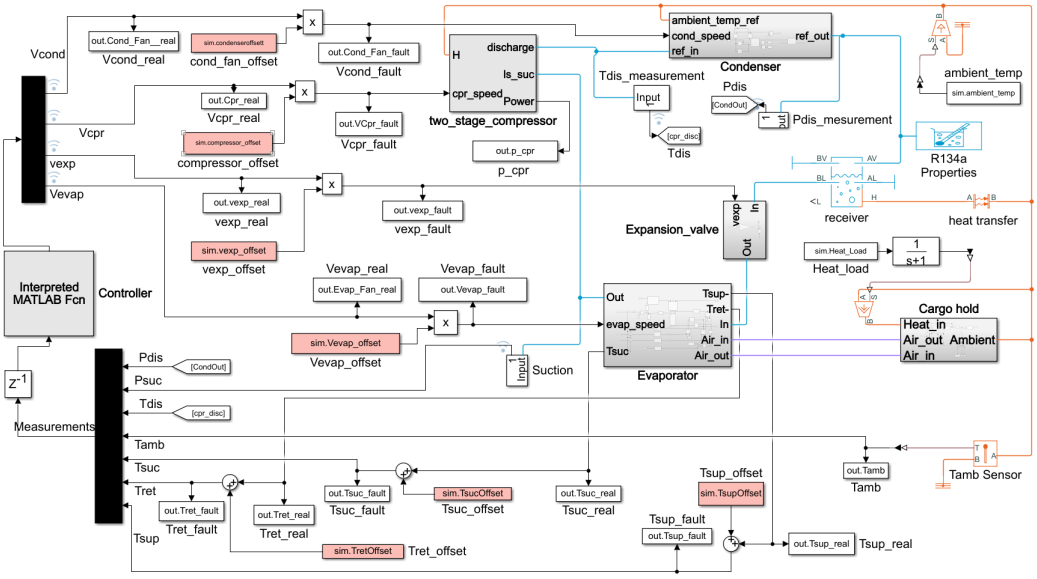

Fig. C.2: The grey blocks indicate the main components of the RS. The red blocks are the faults or offsets that can be applied to each variable.

Therefore, twenty-one classes are studied, including non-faulty condition. In particular, the expansion valve faults are modeled as wrong valve positions compared to the command signal. In fault 8, the actual valve position is 120 % of the command signal, while in fault 18, the valve opens 80% of the command signal. Fig. C.4 represents four examples of data sets taken from the same model and under the same conditions. These examples represent a non-faulty condition, a suction pressure sensor fault with 0.2 bar positive offset indicating fault 6, a loose expansion valve fault where it reacts 20% more than the commanded value from the controller indicating fault 8, and a blocked expansion valve that reacts 20% less than the commanded value. During data acquisition, the model is run in non-faulty condition until sample 6000. Then, each fault is introduced from sample 6001 to 12000 as seen in Fig. C.4. It is observed that in some cases, such as fault 8, the data looks very similar to some of the other faulty or non-faulty data. Changes in condensing temperature (T_C) is compensated by condenser fan work, because there is a feedback control on condenser fan to keep constant pressure relative to T_{amb} and the controller controls V_{cpr} based on T_{sup}. Thus, it is hard to observe any visual changes in the data characteristics during steady state response. However, in some other cases, the fault affects the controller response immediately, and the changes can be observed in the data easily. For example, fault 6, which is shown in Fig. C.4, clearly

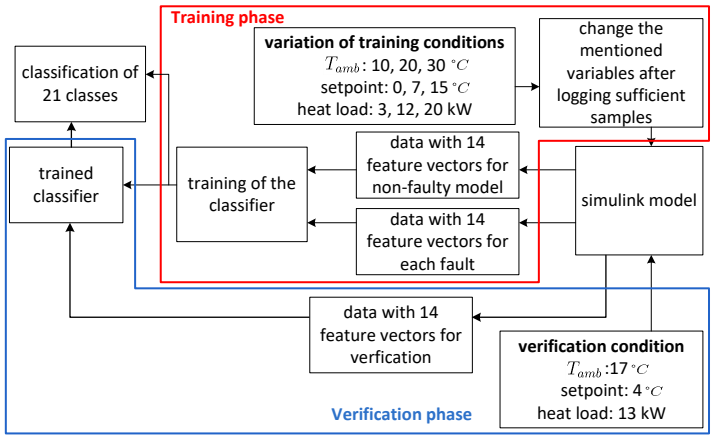

Fig. C.3: An overview of data collection and ML setup. The red section indicates the training phase where data is collected and used for training of the ML model. The blue section shows verification data specification and classification.

gives rise to variations in T_{dis}, and V_{cpr}. The compressor works based on the controller command. In the case of fault 6, ρ and/or P_{suc} which are fed into the controller are measurements of the faulty sensor. Therefore, the compressor behavior is based on the faulty sensor measurement. However, as the real P_{suc} is less than required, it causes drop in mass flow rate. In Fig. C.2, the P_{suc} offset is applied only to the sensor reading. The controller controls both expansion valve opening degree and compressor speed to reach a desired pressure, and when the reading is positively offset the controller must lower the actual suction pressure to reach the desired reading.

2.2 Data specification and dimensionality reduction

The idea behind dimensionality reduction techniques is to remove dependent and redundant features from original data by projecting data to a lower-dimensional space, which holds only essential information. These approaches deal with noisy data and reduce the computation load for classification purposes [13]. In this work, the input data has 14 feature vectors or dimensions, including sensor signals, and some of the variables from RS controller, including superheat temperature, saturated evaporation temperature, compressor cooling capacity/speed, condenser fan speed, and vapour density. Statistical approaches such as PCA and LDA are used to reduce the input data dimensions before passing them through the classifiers. In this paper, all transient part of the data is removed, both for training and validation data. The 14-dimensional data

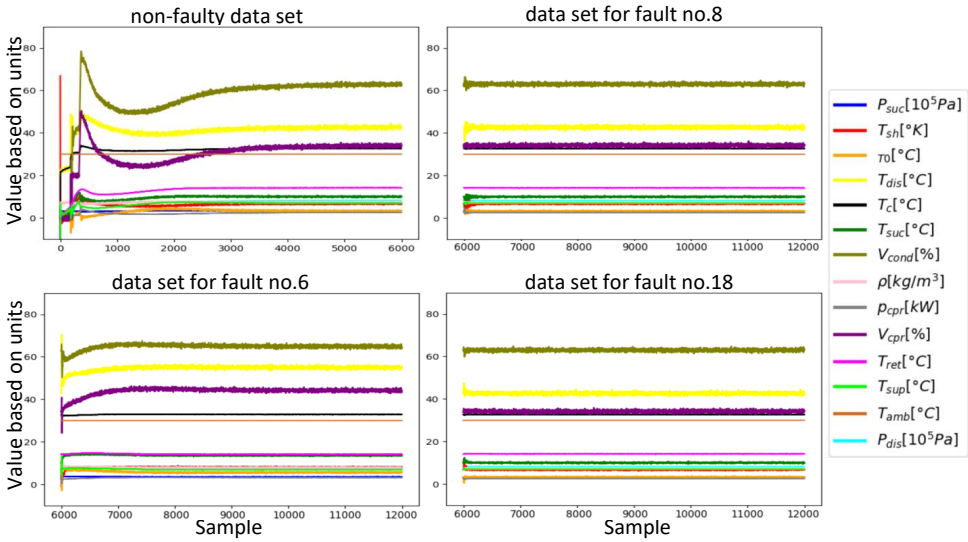

Fig. C.4: Four examples of data set from different classes which have the same system configuration. The set point to $T_{sup} = T_{room}$ is set to 7 $°C$, heat load in the cooling room is 13 kW at the ambient temperature of 25 $°C$.

is reduced to 2-dimensional data using PCA as the input to the SVM. LDA is also used for dimensionality reduction and transfers the data into a 6-dimensional data set before sending the data into the SVM classifier. Moreover, CNN and SVM are also applied to the 14-dimensional data set. For SVM and LDA classification, each class of data contains 1200 samples, and for the CNN classifier, 18000 samples with a sample rate of 1 Hz. Remark that LDA and SVM are shallow learning neural networks which, as an advantage, do not require as many samples as CNN. Too many samples result in too high computation load and low classification accuracy. As described in 2.1, the training data of each class contains various RS operating conditions. These varieties prevent overfitting and increase the model's capability for the classification of unseen operating conditions.

3 Methods

SVM, LDA, and CNN are all supervised learning methods which are sub-fields of the linear classifiers [1]. Supervised ML classifiers categorize a new data set using a pre-trained model. Thus, the model is first trained using input data and defined labels.

CNN is a deep learning classifier commonly used for image processing purposes. A CNN is comprised of two phases of feature extraction and classification. The input data consists of feature vectors $\chi_\phi \in \mathbb{R}^{n\times 1}, \quad \phi = 1,\dots,c$ which are gathered in data matrices $X_\kappa \in \mathbb{R}^{n\times c}$, one for each class $\kappa, \quad \kappa = 1,\dots,\nu$. The numbers $n = n_\kappa$ and c quantify the number of samples in each class and the number of features, respectively. For convenience, it is assumed that all the data matrices have the same dimensions, although this is not a strict requirement.

In the feature extraction phase, the CNN makes use of so-called *neurons* which take data matrices X_κ as input and return (neuron) output $y_\kappa^k \in \mathbb{R}^{n'\times c'}, \quad k = 1,\dots,S$, where S is the number of neurons (see Fig. C.5). Each neuron has a *weight matrix* $W^k \in \mathbb{R}^{\bar{n}\times\bar{c}}$ and a bias matrix $b^k \in \mathbb{R}^{\bar{n}\times\bar{c}}$ associated with it. For each κ, X_κ is partitioned into $n'c'$ (possibly overlapping) submatrices $(x_\kappa)_{ij} \in \mathbb{R}^{\bar{n}\times\bar{c}}, i = 1,\dots,n', j = 1,\dots,c'$. Then the neuron output y_κ^k is a matrix whose entries are defined as:

$$(y_\kappa^k)_{ij} = f(\mathbf{1}^T(W^k \odot (x_\kappa)_{ij} - b^k)\mathbf{1}) \tag{C.1}$$

where $\odot$ denote element-wise multiplication of matrices, $\mathbf{1}$ denotes a vector of ones, and $f : \mathbb{R} \to \mathbb{R}$ is an activation function.

It is noted that the size $\bar{n}\times\bar{c}$ and number S of W^k's are hyper-parameters, which can be tuned during the design of the CNN model to optimally filter different information of the input.

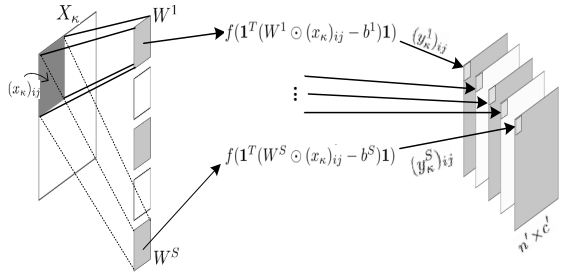

Fig. C.5: A feature extraction layer of CNN, a sub-matrix $(x_\kappa)_{ij}$ is convolved with each weight matrix W^k, resulting in a number of matrices as the output of the layer.

As illustrated in Fig. C.6, the output of the feature extraction phase contains the essential information of the input data. This output is then vectorized as $Y^0 = \mathrm{col}[y_\kappa^k] \in \mathbb{R}^{n'c'S\nu\times 1}$ before being used as input to the classification phase, which is a fully connected Multi-layer Perceptron, see [15], and [16] with N_{MLP} fully connected layers. The output vector of each MLP layer $Y^l \in \mathbb{R}^{n_l\times 1}$ is computed recursively as

$$Y^l = \hat{f}(W_l\ Y^{l-1} + b_l), \qquad l = 1,\dots,N_{\mathrm{MLP}} \tag{C.2}$$

where $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$ is a layer weight matrix, $b_l \in \mathbb{R}^{n_l \times 1}$ is a bias vector, $\hat{f} : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ is the l'th layer's neuron activation function, and $n_{N_{MLP}} = \nu$.

The output $\hat{y} \in \mathbb{R}^{\nu}$ of the CNN is generated by the so-called *Softmax* activation function where the κth coordinate of $\hat{y}$ is given by:

$$\hat{y}_\kappa = \frac{\exp\left(Y_\kappa^{N_{\text{MLP}}}\right)}{\sum_{j=1}^{\nu} \exp\left(Y_j^{N_{\text{MLP}}}\right)} \tag{C.3}$$

with $Y_\kappa^{N_{\text{MLP}}}$ being the κth coordinate of $Y^{N_{\text{MLP}}}$.

Here, it is noted that since the CNN output is normalized ($\sum_{\kappa=1}^{\nu} \hat{y}_\kappa = 1$), $\hat{y}_\kappa$ may be considered as the probability of a new input X belonging to class κ.

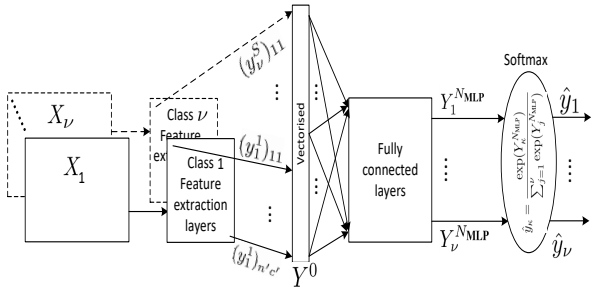

Fig. C.6: General CNN structure for ν classes.

During the training process, the estimation of the classes are compared with the true labels y_κ using a loss function. The loss function is also a hyper parameter that needs to be determined for the model; a common loss function is *cross entropy*:

$$L = -\sum_{\kappa=1}^{\nu} y_\kappa \ \ln(\hat{y}_\kappa). \tag{C.4}$$

The training process aims at adjusting the weights in such a way that better prediction of the correct class is achieved. In other words, the minimum loss is obtained.

Minimization of the loss function can be done using different optimization techniques; the most common being *Backpropagation* [16], which is a variant of gradient descent. Once the weights have been adjusted to yield the optimal output for a validation data set, this model can be used to classify unlabeled, new data.

3.1 LDA classifier

Linear discriminant analysis (LDA) can be used both for dimensionality reduction and classification purposes. In LDA, as it is depicted in Fig. C.7, linear separation of classes is done after projecting data onto another space. LDA seeks a large separation between transformed classes compared to the original one after the dimension of the transformed data is reduced. A transformation matrix is obtained by use of the between-classes variance and the variance within each class [16].

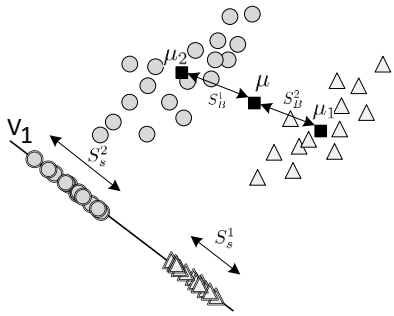

Fig. C.7: LDA visualisation for dimensionality reduction from two to one-dimensional space.

The variance between classes $S_B \in \mathbb{R}^{c\times c}$ is calculated as follows:

$$S_B = \sum_{\kappa=1}^{\nu} (\mu_\kappa - \mu)^T (\mu_\kappa - \mu) \tag{C.5}$$

where $\mu_\kappa \in \mathbb{R}^{1\times c}$ is the mean value of class κ, and $\mu \in \mathbb{R}^{1\times c}$ is mean of all μ_κ. Afterwards, the within-class variance $S_s \in \mathbb{R}^{c\times c}$ is calculated by

$$S_s = \sum_{\kappa=1}^{\nu} \sum_{j=1}^{n} ((X_\kappa)_j - \mu_\kappa)^T ((X_\kappa)_j - \mu_\kappa) \tag{C.6}$$

where $(X_\kappa)_j$ is the jth row (or sample) in X_κ.

S_s and S_B are used to find the transformation matrix $\Omega \in \mathbb{R}^{c\times c}$ defined as

$$\Omega = S_s^{-1}\, S_B \tag{C.7}$$

Afterwards, this transformation matrix is used to generate data in another space in which the classes are linearly separable. In order to reduce the dimensions of the data in the new space, eigenvectors and eigenvalues of Ω are obtained. The eigenvectors

with higher eigenvalues carry more information of the data distribution [17]. Order the eigenvalues of Ω in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_c$, choose the first $\alpha \leq c$ corresponding eigenvectors and organize them in a new matrix $V = [v_1\, v_2 \cdots v_\alpha] \in \mathbb{R}^{c\times\alpha}$. The lower-dimensional samples $r_j \in \mathbb{R}^{1\times\alpha},\ j = 1, \ldots, n$ in class κ are then the rows of the matrix product $X_\kappa V$.

3.2 SVM classifier

Support vector machine (SVM) is a supervised machine learning method and linear classifier which classifies data into two or more classes. In the sequel we focus on the case of two classes.

Consider the two classes $X_\kappa,\ \kappa = 1, 2$ containing the samples as rows and set $y_j = -1$ or $y_j = 1$ if $x_j \in \mathbb{R}^{1\times c}$ is a row in X_1 or a row in X_2, respectively. Assume that the two classes are linearly separable, that is, the samples of each class can be separated by a (linear) hyper plane. Then there exists a hyper plane

$$H = \{x \in \mathbb{R}^c \mid xw^T + b = 0\}$$

with weight $w \in \mathbb{R}^{1\times c}$ and bias $b \in \mathbb{R}$ such that $1/||w||$ is the distance from H to the nearest sample in class 1 and class 2. These nearest samples are usually called support vectors (see Fig. C.8). Moreover, w and b may be found as the solution to the optimisation problem

$$\min_{w,b}\ \frac{1}{2}\|w\|^2 \tag{C.8a}$$

s.t.

$$y_j(x_j w^T + b) \geq 1, \quad j = 1, \ldots, n \tag{C.8b}$$

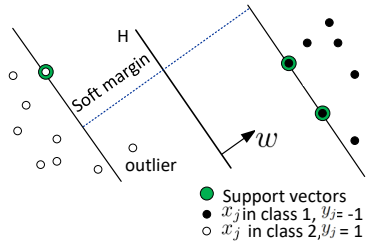

Fig. C.8: Finding a classification hyperplane.

The optimal (or hard) margin (that is, $1/||w||$ with w the solution to (C.8)) may not always lead to the best result when feeding unseen data to the model. The optimal

margin might result in overfitting or margin violations. In particular, outliers can fall into the wrong class and be misclassified [18]. In practice, the classifier is allowed to do small misclassifications during the training, which is called *soft margin* (shown in Fig. C.8). To do so, a slack variable ζ is added to the optimization problems:

$$\min_{w,b,\zeta} \ \frac{1}{2}\|w\|^2 + C\sum_{j=1}^{n}\zeta_j \tag{C.9a}$$

s.t.

$$y_j(x_j w^T + b) \geq 1 - \zeta_j \tag{C.9b}$$

$$\zeta_j \geq 0, \quad j = 1, \ldots, n \tag{C.9c}$$

where C is a hyper parameter that determines the size of the allowed misclassification. The size of the parameter C is tuned by software such that the classification accuracy of unseen data is high.

In many classification problems a linear classification is not possible. The kernel trick is a method for dealing with this case. It yields a transformation of the input space, that is the space which the samples belong to, into another higher dimensional space, in which the samples are linearly separable [18]. This new space is typically called the feature space. The kernel trick relies on the use of kernel function. In this work we consider a special class of kernel function, called the Gaussian Radial Basis Functions (GRBF) given by

$$\Phi(x, x') = \exp\left(-\gamma \parallel x - x' \parallel^2\right) \tag{C.10}$$

The hyper parameter $\gamma > 0$ determines the influence of each sample on selecting the hyper plane during training. It should be noted that choosing γ too big results in overfitting and choosing γ too small leads to under-fitting of the model [16].

3.3 Multi-class classification

In the case of more than two classes, the problem can be solved using two approaches. The first one is to consider each class against the rest of the classes and is called One Versus the Rest (OVR). For the model training using OVR, one binary classifier is used for each class against all the other classes as the second category. Therefore, for a data set including ν classes, ν binary classifiers are created. For unseen data classification, each classifier is tested to determine to which class the new sample belongs. However, in many cases, the result of OVR is inconsistent as the sample can belong to either more than one class or none of them, illustrated as the gray stars in Fig. C.9. Since, OVR picks one class against all other classes together, the number of samples in the corresponding class is typically a lot fewer than the rest of the classes. Therefore, the big difference between the number of samples often impacts the decision boundary [16].

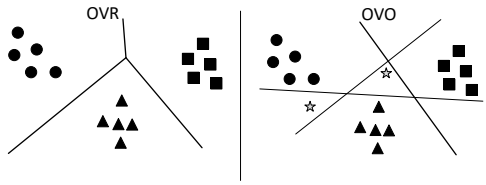

Fig. C.9: multi-class data classification using OVR at the left and OVO at the right.

The second multi-class classification approach takes each class versus another and is called the One Versus One (OVO) approach. Thus, for each pair of classes, one classifier is trained. Finally, $\frac{\nu(\nu-1)}{2}$ classifiers determine each class boundaries as shown in Fig. C.9.

The OVO approach is not as computationally effective as OVR due to using more classifiers. Moreover, the OVO approach has a tendency to overfit [19]. However, in the end, a certain amount of trial and error is unavoidable in selecting a multi-class SVM classifier, as it depends on the input data and feature space.

4 Experiments

In this work, PCA and LDA are built in Python for dimensionality reduction purposes. It is advantageous to use lower-dimensional input data if it reduces the computation time of the classification and/or increases accuracy by removing redundant information in the data set such as noise, etc. This work tests and compares PCA-SVM and LDA-SVM models to the SVM classifier with full-dimensional data. The algorithms are built using the `scikit-learn` library in python which provides many efficient algorithms in ML, dimensionality reduction and classification. In [20], the ways of implementing aforementioned ML techniques in the `scikit-learn` library are described. In this work, the label -1 is assigned to non-faulty data, while other labels are specified in Table C.1. Moreover, the classifiers are fed with two sets of training data which are described in section 2, in order to evaluate the qualification of the training data.

4.1 Full-dimensional classifiers

The input data used for the SVM model includes $n = 1200$ samples of 14 feature vectors for each class. In addition, the input data contains samples from different system configurations. Each sample is labelled with one of the labels in Table C.1. The SVM classifier performs OVO classification using C = 1000, and $\gamma = 0.01$ (see section 3.2); the hyperparameters were found by trial-and-error. The result of classification is

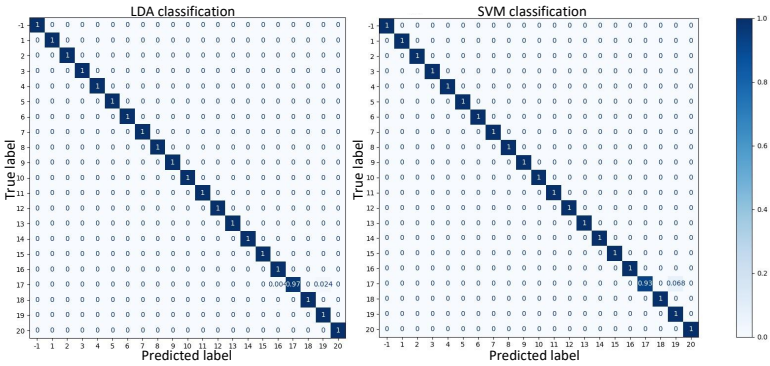

Fig. C.10: LDA and SVM response for classification of 21 classes.

represented in Fig. C.10. *True labels* are the labels assigned to each class during the training phase, while *Predicted labels* refers to the prediction of the classifier during the training process. Thus, the diagonal values represent correct classifications. In this test, 250 samples with 1 Hz sample rate are selected for prediction. The SVM result shows high classification accuracy for most of the classes, and there are no false positives. At 93% accuracy, the broken compressor with label 17 in Table C.1 is the only fault that is misclassified.

As mentioned in section 3, LDA can be used both for dimensionality reduction and classification purposes. Here, LDA is used to classify all 21 classes of data while reducing the dimensions of the input data from 14 to 5. As shown in Fig. C.10, the response of the LDA classifier is very similar to SVM classification, exhibiting 100% classification accuracy for most of the classes and no false positives. The only misclassification of about 3% is the broken compressor, which is mistaken for either P_{suc} sensor negative offset or broken evaporator fan.

CNN is a deep learning model and needs more samples compared to LDA or SVM. In the CNN model experiment, the data set for each class contains 12000 samples of all 14 feature vectors. The classification response of the training is represented in Fig. C.11. The CNN classifier obtained a total accuracy of 94% and could classify most of the faults with 100% accuracy. The noticeable drawback is the false positive rate of 58%. The non-faulty condition was misclassified as classes with labels 8 and 18, which are both expansion valve malfunctions.

4.2 Reduced-dimension classifiers

In this part, PCA and LDA are used to reduce the input dimensionality. These approaches are investigated to see whether PCA or LDA can improve classification results.

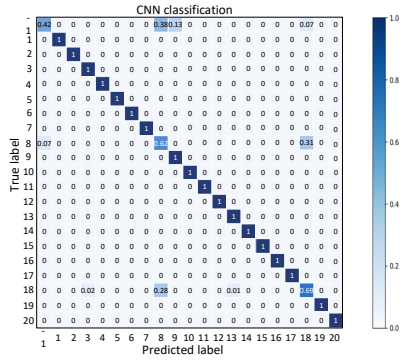

Fig. C.11: Training response of the CNN classifier.

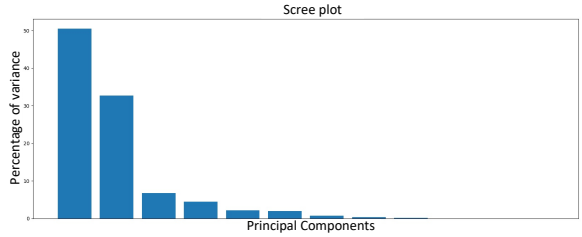

Fig. C.12: The first two principle components contain the most variation among all 14 principle components.

In addition, it is vital to study whether low dimensional inputs can reduce training computation time in the case of PCA and LDA.

After feeding data into PCA and transforming to the new space, it appears that the first two dimensions of the transformed data contain more than 80% of the variations in the new space, as seen in Fig. C.12. Therefore, the first two principal components are used as the inputs to the SVM instead of 14-dimensional data. Fig. C.13 shows the response of the PCA-SVM classifier with $C = 1000, \gamma = 0.01$, and OVO decision function.

The result of PCA-SVM shows misclassification of most of the classes. PCA causes classes to overlap as the most uncorrelated information is squeezed into the first two principal components. The result of PCA-SVM classification is not satisfactory for the multi-class classification even though it represents satisfactory results for binary classification in [13].

LDA is already used for classification, as shown in Fig. C.13. However, it can also be used only for dimensionality reduction; then, the transformed lower dimensional data is used in a classification algorithm such as SVM. The first five eigenvectors corresponding to the first highest eigenvalues indicate that LDA reduces the input dimensions from eleven to five. A LDA-SVM classifier is built using $C = 1000, \gamma = 0.01$, and OVO decision function for the SVM part. The LDA-SVM classifier performs satisfactorily for many of the classes shown in Fig. C.13. However, the model cannot easily distinguish the Non-faulty, loose expansion valve fault, and blocked expansion valve fault. Therefore, these three classes have low accuracy and a false positive rate of 18%, which is not a satisfactory result.

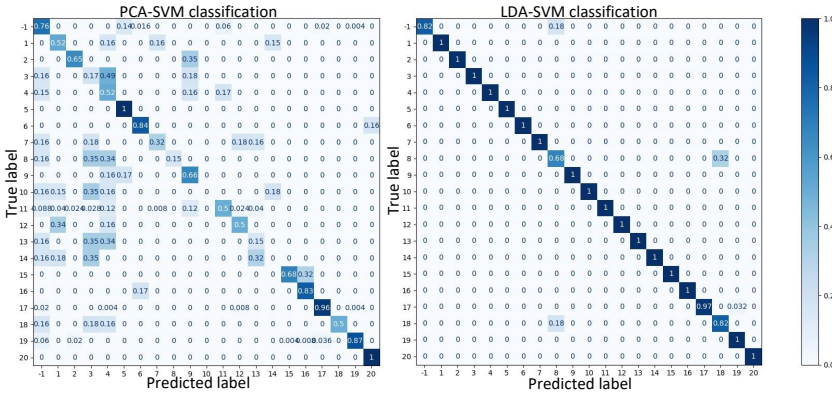

Fig. C.13: PCA-SVM and LDA-SVM response for classifying 21 classes.

4.3 Model comparison

In this part, all five models described above are compared regarding input dimensions, total accuracy, false positive rate, training time, and prediction time. Training time is the computation time a model requires to be trained, including the time for dimensionality reduction in the cases where the input dimension is reduced. The prediction time is the computation time that a trained model uses to classify a test data set. Here, test data and training data are collected from the same system configuration and operation. The results in Table C.2 represents the performance of the training process.

As seen in Table C.2, SVM and LDA achieved the best results, with high accuracy and no false positives. However, the prediction time is relatively low for the LDA classifier compared to SVM, PCA-SVM, and LDA-SVM. On the other hand, the CNN classifier has the lowest prediction time, but the false positive rate is unacceptable.

Table C.2: Comparison of different classifiers

model	dimensions	accuracy	false positive	training time	prediction time
SVM	14	99.6%	0%	1.1 s	1 s
LDA	14 to 5	99.8%	0%	3.2 s	0.3 s
CNN	14	94%	68%	112.5 s	0.1 s
PCA-SVM	14 to 2	55.4%	24%	7.2 s	5.6 s
LDA-SVM	14 to 5	96.6%	18%	1 s	1.1 s

Table C.3: Robustness of classifiers against different operating conditions

model	accuracy	false positive	prediction time
SVM	76%	92%	3.1 s
LDA	52%	100%	0.2 s
LDA-SVM	57%	100%	2.1 s

Therefore, LDA is found as the best model for multi-fault classification. Afterwards, more investigation is done on SVM, LDA and LDA-SVM, which perform better during the training phase.

4.4 The classifiers verification

In this part, the validation data is specified with a set point, heat load and ambient temperature which is different from what are used for the training set. In this data set, T_{set} is 4 °C, heat load is 13 kW and ambient temperature is 17 °C. Fig. C.14 shows the response of SVM, LDA, and LDA-SVM classifier trained with the first training data set, with variations in set point and heat load. The overview of the results in Table C.3 indicates that even though the classifiers did a good job during the training and test, they can not deal with the new data which are taken from a system in a new operating condition. Therefore, the classification results are not satisfactory, especially when looking at the false positive rate.

4.5 Effect of data variation

To deal with the challenge of misclassification of unseen data, a new training data set is fed into the same model, which contains more excitation by varying the RS operation

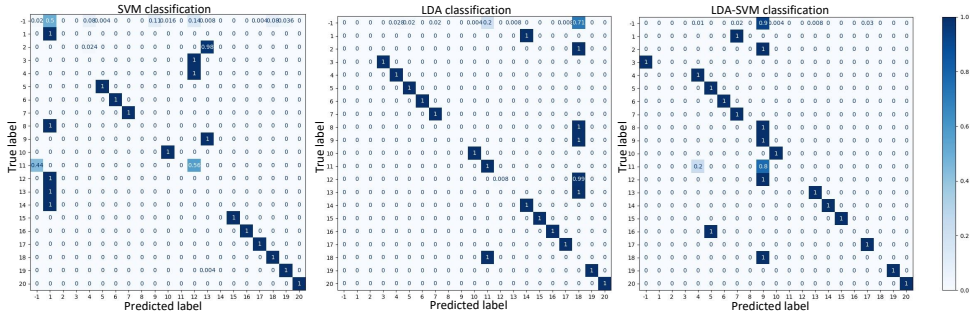

Fig. C.14: Three classification responses of validation data with different system operating condition.

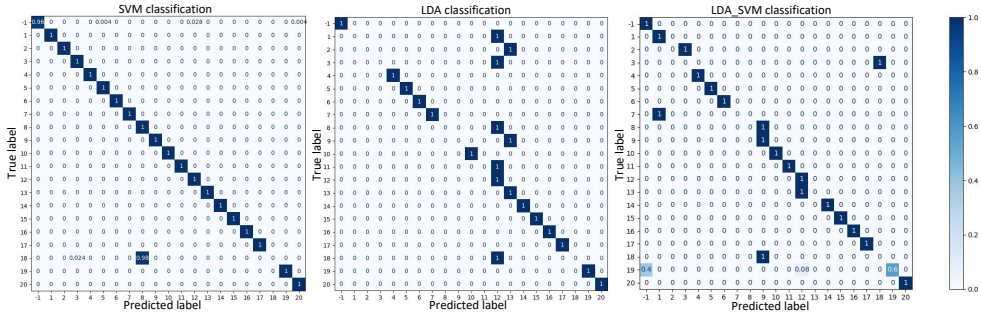

Fig. C.15: Higher classification accuracy after training with new training data for all three classifiers comparing to Fig. C.14 .

around ambient temperature from 10 to 30 °C, set point from 0 to 12 °C, and heat load from 3 to 18 kW. In addition, to obtain better results, all 14 feature vectors are tested to see if one can affect misclassification. Thus, three features of input data, namely, P_{suc}, compressor power consumption and density that were already used, are removed from the training and validation data set as they adversely affect the classification accuracy. The results are depicted in Fig. C.15. The overview of the results in Table C.4 shows that the SVM classifier obtains more accurate results after training with more excited training data and removing the three mentioned feature vectors. However, for the LDA-SVM and LDA classifiers, the most accurate results are obtained when just the power consumption of the compressor and density are removed. Using this adjustment, the false positive percentage is improved a lot and SVM stands alone regarding the diagnosis of all faults simultaneously with high accuracy.

Table C.4: Robustness of classifiers after using qualified training data

model	accuracy	false positive	prediction time
SVM	95%	4%	8.2 s
LDA	66%	0	0.2 s
LDA-SVM	69%	0%	9.1 s

5 Conclusion

In this work, different classifiers are compared to diagnose twenty types of faults simultaneously and non-faulty condition in the industrial RS. The training data is taken from a simulation model which has been used in the development of system control in Bitzer. First, five classifiers, namely CNN, SVM, LDA, LDA-SVM, and PCA-SVM are compared. The training data contains information about the different systems operating through variation of set point and heat load of the cooling room. The test results show that CNN and PCA-SVM do not satisfactorily diagnose the faults. In addition, SVM, LDA, and LDA-SVM can not properly deal with the verification data set taken from different system operations than training data. On the other hand, training data with more excitation can help the classification results when the new training data contains the variations in ambient temperature, set point, and heat load. LDA and LDA-SVM are improved regarding false positive rate to 0%, but these classifiers do not provide the satisfactory classification results of the other classes.

It is seen that SVM has the highest classification accuracy of 95% with a 4% false positive rate. The only class which SVM does not diagnose is the blocked expansion valve, which is misclassified with the loose expansion valve. Therefore, even though this fault is misclassified, we can still trust that the malfunctioning valve needs to be checked by the technicians.

From an industrial point of view, it is very beneficial to have one classifier that can diagnose twenty one classes. Moreover, the classifiers considered in this work can be trained off-line. Off-line training may have two advantages. First, It is possible to train the classifier with simulation data and use the trained classifier for classification of real data to ensure that we do not train the classifier with the real data which are wrongly labeled. Second, the trained classifier would be computationally lighter compared if the training process were to be executed on embedded software as well. This is an advantage when the capacity of the processor of typical refrigeration systems is considered. The SVM model obtained the best classification accuracy at the algorithms tested. If a lower false positive percentage is considered, LDA can be used with a 0% false positive rate only for distinguishing the non-faulty class from the other faulty classes. Therefore, the system could benefit from having two classifiers, to make the diagnosis result more reliable. Before implementation of the classifier on real refrigeration systems, verification

of the trained classifier by using real data from the field will be done in the future work.

Acknowledgment

This work is funded by Innovation fund Denmark and Bitzer Electronics A/S. [fund number:9065-00010B]

References

[1] R. Saravanan and P. Sujatha, *A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification*, 2018, pp. 945–949.

[2] G. Gauglitz, "Artificial vs. human intelligence in analytics: Do computers outperform analytical chemists?" *Analytical and bioanalytical chemistry*, vol. 411, pp. 5631–5632, 2019.

[3] M. Rahnama, A. Vahedi, A. M. Alikhani, and A. Montazeri, "Machine-learning approach for fault detection in brushless synchronous generator using vibration signals," *IET science, measurement & technology*, vol. 13, pp. 852–861, 2019.

[4] M. Quiñones-Grueiro, M. Ares Milián, M. Sánchez Rivero, A. J. Silva Neto, and O. Llanes-Santiago, "Robust leak localization in water distribution networks using computational intelligence," *Neurocomputing (Amsterdam)*, vol. 438, pp. 195–208, 2021.

[5] T. Liu, Y. Fu, X. Xu, and W. Yan, "A cross-layer fault propagation analysis method for edge intelligence systems deployed with dnns," *Journal of systems architecture*, vol. 116, pp. 102 057–, 2021.

[6] J. H. Choi, C. Park, P. Cheetham, C. H. Kim, S. Pamidi, and L. Graber, "Detection of series faults in high-temperature superconducting dc power cables using machine learning," *IEEE transactions on applied superconductivity*, vol. 31, no. 5, pp. 1–9, 2021.

[7] M. Hajji, M.-F. Harkat, A. Kouadri, K. , Abodayeh, M. Mansouri, H. Nounou, and M. Nounou, "Multivariate feature extraction based supervised machine learning for fault detection and diagnosis in photovoltaic systems," *European Journal of Control*, vol. 59, pp. 313–321, 2021.

[8] J. Sana Ullah, D. L. Young, and S. K. In, "A distributed sensor-fault detection and diagnosis framework using machine learning," *Information Sciences*, vol. 547, pp. 777–796, 2021.

[9] M. Shioya, Y. Masukawa, T. Yairi, and K. Yoshida, "Energy fault detection in office building system by machine learning methods," *ASHRAE transactions*, vol. 121, no. 1, pp. 185–, 2015.

[10] A. Choudhary, T. Mian, and S. Fatima, "Convolutional neural network based bearing fault diagnosis of rotating machine using thermal images," *Measurement : journal of the International Measurement Confederation*, vol. 176, pp. 109 196–, 2021.

[11] N. G. Lo, J.-M. Flaus, and O. Adrot, "Review of machine learning approaches in fault diagnosis applied to iot systems," in *2019 International Conference on Control, Automation and Diagnosis (ICCAD)*, 2019, pp. 1–6.

[12] Z. Soltani, K. K. Soerensen, J. Leth, and J. D. Bendtsen, "Fault detection of supermarket refrigeration systems using convolutional neural network," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 231–238.

[13] Z. Soltani, K. K. Sørensen, J. Leth, and J. D. Bendtsen, "Robustness analysis of pca-svm model used for fault detection in supermarket refrigeration systems," in *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE, 2021, pp. 1–6.

[14] 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.

[15] P. S. Geidarov, "Clearly defined architectures of neural networks and multilayer perceptron," *Optical memory & neural networks*, vol. 26, no. 1, pp. 62–76, 2017.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[17] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Communications*, vol. 30, pp. 169–190,, 05 2017.

[18] M. N. Murty and R. Raghava, *Support Vector Machines and Perceptrons: Learning, Optimization, Classification, and Application to Social Networks*, ser. SpringerBriefs in computer science. Springer International Publishing AG, 2016.

[19] J. C. Platt, N. Cristianini, and J. Shawe-taylor, "Large margin dags for multiclass classification," pp. 547–553, 2000.

[20] G. Aurélien, *Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd edition.* O'Reilly Media, Inc, 2019.

Appendix

1 Corrections in Paper C

This appendix is referenced to Paper C. Here, some sections of Paper C are represented, including the important corrections which are distinguished after the publication. The corrections are included in Subsections 2.2, 4.5 and consequently, in Abstract, Introduction and Conclusion. Fig. C.15 and Table. C.4 are replaced with Fig. 16 and Table. 5 in this appendix. These corrections are notified to the publisher for further edits in the article.

Data collection is conducted to record data for each class, including 20 classes of faulty systems and one non-faulty system. After the publication of this work, we observed that in the non-faulty data set, two feature vectors of ambient temperature and setpoint temperature in Subsection 4.5 are swapped unintentionally compared to the data sets for the other 20 classes. This mistake leads to recording non-faulty data in a different space than the other classes and makes the non-faulty data have unique characteristics compared to all faulty data sets. Therefore, it affects the results. Therefore, the data is manipulated in the right order and the experiments of Paper C, Section 4.5. are repeated. The corrections in Paper C are replaced by red marks as follows.

1.1 Corrections "Abstract"

The functionality of industrial refrigeration systems is important for environment-friendly companies and organizations since faulty systems can impact human health by lowering food quality, causing pollution, and even leading to increased global warming. Therefore, in this industry, there is a high demand among manufacturers for early and automatic fault diagnosis. In this paper, different machine learning classifiers are tested to find the best solution for diagnosing twenty faults possibly encountered in such systems. All sensor faults and some relevant component faults are simulated in a high-fidelity Matlab/Simscape model of the system, which has previously been used for controller development and verification. In this work, Convolutional Neural Networks, Support Vector Machines (SVM), Principal Components Analysis SVM, Linear Discriminant Analysis-SVM, and Linear Discriminant Analysis classifiers are compared. The results indicate that the fault detection reliability of the algorithms highly depends on how well the training data covers the operation regime. Furthermore, it is found that a well-trained LDA-SVM can simultaneously classify 18 types of faults out of 20 with 100% accuracy when the verification data is taken from different system configurations. The overall accuracy of this classifier for 21 classes is 86%.

1.2 Corrections "Introduction"

Machine Learning (ML) is a common term for many processing methods used for data-driven tasks. The main intention of ML is to enable computers to learn, predict, or

decide on an unseen data without human assistance [1]. In the 2010s, rapid development of processors, IoT, and an increasing amount of generated data paved the way for large improvements in ML capabilities. Thus, the popularity of ML increased exponentially in many industries. Machine learning is used in various contexts, such as computer vision, text classification, fault detection, language processing, image recognition, and so forth.

The idea of using ML for fault detection and diagnosis dates back to the 1980s where the existing ML methods were not as efficient as specialized experts. However, the technologies have been improved, and as of today, the availability of powerful programming tools and algorithms for self-learning allow computers to make strategic decisions and even diagnose new events [2].

In particular, ML-based methods have been studied for fault detection and diagnosis (FDD) in different fields with promising results. For instance, ML is used for fault detection in brushless synchronous generators in [3], in water distribution network [4], in age intelligence systems [5], and in high-temperature super conducting DC power cables [6]. In [7], several supervised ML algorithms are compared for FDD in photovoltaic systems. In [7], data from non-faulty condition and five different faulty conditions are used both for training and test; and the results confirm that supervised learning algorithms can be used for fault detection and ease the FDD procedure. Moreover, machine learning models are compared for sensor fault detection in [8], in which five types of sensor faults are emulated, namely, drift, bias, precision degradation, spike, and stuck faults.

For fault detection in office building systems, various data mining methods, in particular, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Kernelized Discriminant Analysis (KDA), semi-supervised LDA, and semi-supervised KDA have been compared in [9]. In [10], different component faults in a rotating machine are classified using a Convolutional Neural Network (CNN) algorithm. According to [11], in many industrial applications, good system models are difficult or even impossible to obtain due to the system's complexity or large numbers of configurations involved in the production process. The refrigeration industry is not an exception, as the system configuration varies based on different owners' demands. Hence, model based FDD is often sensitive to model parameters in such a way that small changes in the system may lead to a poor fault detection response. In such cases, ML can be a viable approach to handling unseen situations when well trained.

In [12], a CNN model is used for evaporator fan fault detection in supermarket refrigeration systems. The same system configuration and information are used in [13] to classify the same fault and investigate the robustness of the fault detection model. However, instead of CNN, shallow learning Support Vector Machines SVM and PCA-SVM classifiers are used. In [14], SVM and PCA-SVM are studied for the detection of 8 type of faults in a simulated vapour-compression refrigeration system in which PCA-SVM achieved a better result compared to SVM and back-propagation neural network.

In the refrigeration industry, good performance of a fault detection algorithm can be defined as high classification accuracy, low computation time, and low false positive rate. High classification accuracy ensures an accurate fault description for the technicians for quick troubleshooting, while low computation time is important because it lowers the detection time and the hardware cost. A low false positive rate increases the reliability of the fault detection model and results in lower expenses regarding service call rate. Therefore, it is essential to evaluate the FDD algorithms based on these factors.

Because of increasing usage of digitalization in refrigeration systems (RS), many companies aim for improving existing FDD performance by utilising various data. As mentioned above, FDD algorithms perform satisfactorily in many other applications; thus, data driven FDD algorithms are selected and evaluated in this work. That is, we evaluate and optimize various FDD algorithms for the purpose of selecting the best classifier for use in RS industry applications.

The main contributions of this study is summarized below:

- A deep learning and several shallow learning classifiers are proposed for detecting and diagnosing twenty types of faults in RS.
- Importance of training data qualification regarding data variation and features selection is illustrated.
- All of the proposed classifiers are compared regarding classification accuracy, computation time and false positive rate.
- The best approach from an industrial perspective is proposed to detect a faulty system and localize the fault.

In this study, all sensor faults and some component faults are simulated using a high-fidelity RS model. The model is already in use at Bitzer Electronics to develop and verify control algorithms. Notice that we will restrict our attention to steady state operating conditions, which are commonly encountered in industrial application such as reefer containers, cold storage houses and so on. It is acknowledged that transient operation is important in many applications as well, e.g., in supermarket refrigeration systems. However, transient behavior presents its own set of unique challenges, and is considered out of scope of this work. The faults include positive and negative offsets in sensors as well as specific component faults; the faults are detailed in section 2. Three classifiers, namely CNN, SVM, and LDA, are compared to diagnose every selected fault. For pre-processing of the input data LDA and PCA are compared. The results indicate that the SVM classifier is the superior method considering the total classification accuracy of the models, being able to diagnose 18 classes out of 21 with 100% classification accuracy, while the non-faulty class is one of the classes that is misclassified with 70% false positive rate. LDA-SVM can also classify 18 classes out of 21 with 100% accuracy, while the non-faulty class is classified with 0% false positive rate, and all three misclassified classes

are from faulty classes. Comparing the LDA classifier with the LDA-SVM and SVM, a general lack in performance is observed when it is used for both dimensionality reduction and classification.

The remainder of this paper are structured as follows. First, refrigeration systems background and specification, as well as data acquisition and its specification, are introduced in section 2. Then, in section 3, the mathematical approaches of the classifiers mentioned above are explained. Afterwards, the specification of each model and the result of the classification is presented in section 4. Finally, the work is concluded in section 10.

1.3 Corrections "Data specification and dimensionality reduction"

The idea behind dimensionality reduction techniques is to remove dependent and redundant features from original data by projecting data to a lower dimensional space, which holds only essential information. These approaches deal with noisy data and reduce the computation load for classification purposes [13]. In this work, two data sets are used. The first input data set has 12 feature vectors or dimensions, including sensor signals and some of the variables from RS controller, including superheat temperature, saturated evaporation temperature, condenser fan speed, compressor power and vapour density. Sections 4.1 to 4.4 covers several experiments using this data. Then in Section 4.5, another dataset with 14 feature vectors is used in which ambient temperature and setpoint are added as 2 new feature vectors. Statistical approaches such as PCA and LDA are used to reduce the input data dimensions before passing them through the classifiers. In this paper, all transient part of the data is removed, both for training and validation data. The original data dimensions are reduced to 2-dimensional data using PCA as the input to the SVM. LDA is also used for dimensionality reduction and transfers the data into a 6-dimensional data set before sending the data into the SVM classifier. Moreover, CNN and SVM are also applied to the 12-dimensional data set. For SVM and LDA classification, each class of data contains 1200 samples, and for the CNN classifier, 18000 samples with a sample rate of 1 Hz. Remark that LDA and SVM are shallow learning neural networks which, as an advantage, do not require as many samples as CNN. Too many samples result in too high computation load and low classification accuracy. As described in 2.1, the training data of each class contains various RS operating conditions. These varieties prevent overfitting and increase the model's capability for the classification of unseen operating conditions.

1.4 Corrections "Effect of data variation"

To deal with the challenge of misclassification of unseen data, a new training data set is fed into the same model, which contains more excitation by varying the RS operation

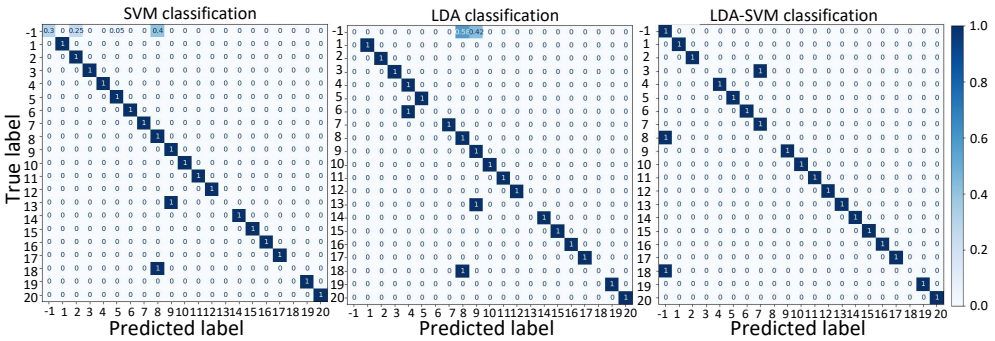

Fig. 16: Higher classification accuracies using validation data for all three classifiers after training with new training data comparing to Fig. C.14 .

around ambient temperature from 10 to 30 °C, set point from 0 to 12 °C, and heat load from 3 to 18 kW. In addition, two new feature vectors, T_{amb} and T_{set}, are added to the input data. To obtain better results, the effect of every feature vector is tested to investigate if one can affect misclassification. Thus, three features of input data, namely T_{amb}, density, and power consumption of the compressor, are removed from the training and validation data sets as input to the three algorithms as they adversely affect the classification accuracy. Finally, the hyperparameters are tuned to optimize model performance. The results are depicted in Fig. C.15, and the optimized models are specified below:

- SVM classifier is specified by OVO function, RBF kernel function, $C = 1000$ and $\gamma = 0.01$. The input to this SVM is 11-dimensional, where T_{amb}, density and power consumption of the compressor are removed from the original 14-dimensional data.
- LDA classifier uses six linear discriminants meaning that the data dimensions are reduced from 11 to 6.
- LDA-SVM classifier is specified by OVO function, RBF kernel function, $C = 1000$ and $\gamma = 0.01$. The input to the LDA-SVM is 11-dimensional, the same as the SVM classifier.

The overview of the results in Table 5 shows that the LDA-SVM classifier obtains more accurate results after training with more excited training data using specified feature vectors mentioned above. LDA-SVM model obtained a 0% false positive rate while SVM classify the non-faulty data with 70% false positive rate, and LDA can classify the non-faulty data with a 20% false positive rate. However, regardless of the non-faulty class, the SVM model can diagnose 18 faults out of 20 with 100% accuracy simultaneously.

Table 5: Comparison of classifiers after training the algorithm using more excited data

model	accuracy	false positive	prediction time
SVM	87%	70%	0.4 s
LDA	81%	100%	0.3 s
LDA-SVM	86%	0%	1.5 s

1.5 Corrections "Conclusion"

In this work, different classifiers are compared to diagnose twenty types of faults simultaneously and non-faulty condition in the industrial RS. The training data is taken from a simulation model which has been used in the development of system control in Bitzer. First, five classifiers, namely CNN, SVM, LDA, LDA-SVM, and PCA-SVM are compared. The training data contains information about the different systems operating through variation of set point and heat load of the cooling room. The test results show that CNN and PCA-SVM do not satisfactorily diagnose the faults. In addition, SVM, LDA, and LDA-SVM cannot properly deal with the verification data set taken from different system operations than training data. On the other hand, training data with more excitations can help the classification results when the new training data contains the variations in ambient temperature, set point, and heat load. In general, the performance of all classifiers is improved using the more excited data. Meanwhile, LDA-SVM is significantly improved regarding false positive rate to 0% and diagnosing 17 types of faults out of 20 with 100% accuracy. It is seen that SVM has the highest classification accuracy of 87% but with a high false positive rate. Regardless of the non-faulty class, only two faulty classes, namely, blocked expansion valve fault and negative offset of return temperature sensor, are misclassified with loose expansion valve fault and evaporator fan low performance, respectively. From an industrial point of view, it is very beneficial to have one classifier that can localize 18 different faults out of 20 in RS. Moreover, the classifiers considered in this work can be trained off-line. Off-line training may have two advantages. First, it is possible to train the classifier with simulation data and use the trained classifier for classification of real data to ensure that we do not train the classifier with the real data which are wrongly labeled. Second, the trained classifier would be computationally lighter compared if the training process were to be executed on embedded software as well. This is an advantage when the capacity of the processor of typical refrigeration systems is considered. The SVM model obtained the most classification accuracy at the algorithms tested. If a lower false positive percentage is considered, LDA-SVM can be used with a 0% false positive rate only for distinguishing the non-faulty class from the other faulty classes. Therefore, the system could benefit from having two classifiers, to make the diagnosis result more reliable. Before implementation of the classifier on real refrigeration systems, verification of the

trained classifier by using real data from the field will be done in the future work.

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-742-0

AALBORG UNIVERSITY PRESS