CHILLER DIAGNOSTICS

Machine learning approach Carrier

Vilius Koegst

Tatiana Orlova

LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6167
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2022

Abstract

Chillers are large and complex machines that are used for temperature regulation in large buildings and plants. An undetected fault in the machine can lead to extended downtime and cause both great financial losses and increased environmental impact. Therefore Carrier, as manufacturer of chillers, is interested in developing fault detection algorithms capable of detecting the faults early so that they can be addressed before complications can occur. The faults, studied in this thesis, are refrigerant leakage, fouling (accumulation of unwanted material on a surface) in the evaporator and fouling in condenser, some of the main components in the machine.

The thesis aims to implement a fault detection and diagnostic algorithm (FDD), using supervised machine learning methods, and provide accurate and reliable identification of chosen faults. FDD is done by creating three binary classifiers, one for each fault, and by training the classifier on a labeled dataset, simulated from a high-fidelity chiller model. Inputs for data generation in the chiller model come from the field data from 35 days in summer. Machine learning models - decision tree, logistic regression, and support vector machine, were used to develop the FDD algorithm and their performance was compared with 4 key indicators: accuracy, false alarm rate, latency, and isolation.

The results have shown that the problem occurred to be linearly separable, with respect to this particular dataset. Support vector machine model has achieved the highest scores in the detection of all three faults with the lowest fault severity level. Refrigerant leakage has proven to be the most difficult fault to detect, partly because the fault severity was almost close to normal operating conditions. Moreover, it was estimated that three signals were needed to detect leakage, while fouling could be separated by only the two most important signals. Finally, the transient data was removed and the training dataset was reduced to 1 day, which proved to be enough to detect and isolate the selected faults.

Acknowledgements

We would like to express our gratitude to the following people, that helped us a lot with the thesis:

Johan Grönqvist, our supervisor at Lund University, for his support and guidance during the entire process of writing the thesis; for his help to plan, organize and structure the work process;

Clas Jacobson, our supervisor at Carrier, for sharing his experience, valuable advices in statistics, for keeping our focus on the main path of the thesis;

Robert Leffler, whose ideas about machine learning methods have inspired us for deeper study research, as well as his clear explanations about chillers, signals and faults helped us to understand physics better;

Bryan Eisenhower, who kindly provided us simulated datasets and always gave a quick response on our needs regarding data generation, who participated a lot in thesis discussions and helped us to learn how chillers modeled and controlled;

Magdalena Atlevi, who we had the pleasure to meet as the first person from Carrier and who helped us to start with the thesis as well as she organized and led the meetings with people from Carrier;

Ramprasad Yelchuru, Mitra Biswajit, Liang Chang, Pavan Mangaiahgari, who have shown much enthusiasm and interest in the thesis during our weekly meetings;

Last but not least, we are grateful for the time that we spent together at Carrier office with the other two Master thesis students, Emil Sundström and Henrik Lindström, for having the opportunity to discuss our thesis and share ideas.

Contents

Acronyms

FDD Fault detection and diagnostics.

FN False negative.

FP False positive.

FPR False positive rate.

HVAC Heating, ventilation and air conditioning.

KPI Key performance indicators.

LMTD Logarithmic mean temperature difference.

LR Logistic regression.

LTD Linear temperature difference.

OAT One at time.

SVM Support vector machine.

TN True negative.

TP True positive.

UA Overall heat transfer coefficient.

List of Figures

List of Tables

1

Introduction

1.1 Motivation

With the growing complexity and performance requirements in industrial processes, more and more advanced automation solutions are implemented in machinery. This increases the efficiency of the machines but at the same time makes them even more complex and expensive. To provide robust and safe operation, early fault detection has significant importance for manufacturers as failure can lead to extended downtime and cause great financial losses.

Carrier, as one of the leading manufacturers and distributors of heating, ventilation, and air conditioning (HVAC) systems, emphasizes their increasing need for fault detection and diagnosis (FDD) of chillers. Currently, faults in the chillers are sometimes not detected on a customer site until they suffer critical failure and shut down. The diagnostic data, available to the technicians, is often lacking, overflowing and/or information poor, which makes the repair process time-consuming and reliant on the technician's competence and experience. Existing threshold algorithms do not capture the broader dynamics of the chillers and have proven to be insufficient to meet Carrier needs for early fault detection.

Considering this, Carrier sees machine learning as one of the tools that can ultimately enhance existing fault diagnostic methods. In this thesis, preliminary research into the development of such diagnostics algorithm is performed. As the manufacturer of the chillers, Carrier has a great understanding of the system dynamics and has created high-fidelity models. These models of chillers can be modified to simulate the system behavior with faults, enabling data generation needed for training and testing of a machine learning algorithm.

The machine learning fault detection method to be developed should have an accuracy of no less than 95%, have a very low false alarm rate and a short time delay, i.e. detect the faults early before the operation of the chiller is significantly impacted. This thesis, being a pre-study of FDD using machine learning, can give a better understanding of product performance in the field, which can help to form future design decisions for a model. Suggestions on useful sensors to add can also increase aftermarket revenue through better service offerings. Finally, the thesis is

supposed to give a good base for future studies of more complex and flexible machine learning algorithms.

1.2 Problem description

In this project, a dataset generated by Carrier will be used to develop an FDD machine learning algorithm. The algorithm has to be interpretable and be able to correctly detect and isolate three different faults, namely fouling in the evaporator, fouling in the condenser and charge loss. An accuracy of at least 95% is required together with a false alarm rate of no more than 5%.

Carrier is interested in answering the following questions:

- Can such an algorithm be implemented using simple machine learning models?
- Can it be done using simulated data from their chiller models?
- What machine learning model is the most suitable for this application?

To limit the scope of the thesis, the broad nature of the questions above needs to be narrowed down. Therefore, the questions that will be answered in this thesis will be specific to the dataset provided. These questions are:

- Can such an algorithm be implemented using simple machine learning models for the provided dataset?
- Which of the three machine learning models: support vector machine, linear regression or decision tree, performs the best?
- What signals are needed for detection and isolation of these selected faults?
- How much training data is needed to meet the requirements?

The ultimate goal of this thesis is to get an insight into what machine learning methods are suitable for this application and to form an understanding of which system variables need to be monitored and what information needs to be extracted from them to be able to successfully detect and isolate those selected faults. This will be done both statistically (with the help of feature importance methods) as well as with physics-informed predictions.

1.3 Thesis Disposition

The thesis is divided into 6 main chapters: introduction, background, methodology, results, discussion and conclusion with suggestions for future work. In the background part, the basic physics and functionality of a chiller are presented as well as

a short introduction to the principles of FDD and machine learning algorithms. The motivation for the chosen faults and a description of the dataset are also included there. The methodology part describes further how the data was preprocessed, split and labeled. The choice of strategy for the implementation of the FDD algorithm as well as its validation is also explained there. In the result section one can find the tables and plots for the performance of the implemented FDD algorithm, results and data uncertainties are then discussed in the next chapter. Finally, the conclusion section has a summary of what was achieved in the thesis as well as suggestions for improvements and future work.

2

Background

2.1 Fault Detection and Diagnostics

A fault is a non-temporary change in process parameters, and it can have many different causes. For example, it can appear because of wear in certain components, unfit operational conditions or some external events. Depending on the process, the location and the severity of the fault, it can result in anything from a slight change in performance or energy consumption to a critical failure and shutdown. Therefore, early detection of faults is imperative for the safe and effective operation of any machine.

The rising amount of sensors and internet integration in industrial systems and machines have made it possible to monitor their performance and health in real-time. This creates an opportunity for the application of automatic detection algorithms that can detect a fault before it significantly disturbs the system's performance, or disable the system during potentially dangerous operating conditions. Such FDD algorithms have become more and more important with the rising complexity of machines, as the control systems that are designed to run the machines often can hide the smaller faults until they become critical.

Different Approaches

FDD can be implemented in a large variety of ways. One of the more common approaches in the industry is a data-driven method of setting warning and alarm thresholds for some chosen system variables. The appropriate levels for the thresholds are set based on historical data, accumulated during normal operation. Although this method is relatively simple, it considers each monitored signal on its own, requires historical data, and also is very static. Other, more complex methods include neural networks, comparisons with a model twin, logic-based rule sets, etc. As with so many other things, there is no explicit best method, and one has to be selected, based on the system, available resources and other limitations at hand.

For this thesis, Carrier has generated both data for normal operation and data with faults, using simulations from their chiller models. The signal data is then

broken down into specific features, that are analyzed and used to train a machine learning algorithm.

Key Performance Indicators

To evaluate the performance of the implemented FDD algorithm, four main Key Performance Indicators (KPIs) are used in this thesis: accuracy, false alarm rate, isolation and latency.

The accuracy of the FDD algorithm is defined as the percentage of correctly identified cases and can be calculated with the following equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2.1}$$

where TP and TN are true positives respectively true negatives. True positive gives a number of correctly identified faults, and true negatives - the number of non-fault cases. FP and FN stand for false positives and negatives, meaning either flagging a healthy system as a faulty one or failing to detect a fault. These measurements are better visualized in form of a table, called a confusion matrix. Elements on the diagonal represent correctly predicted classes, and elements on off-diagonals represent mislabeled cases.

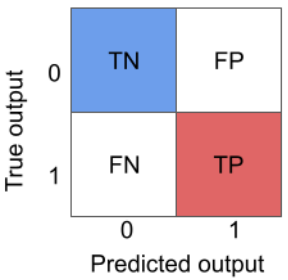

Figure 2.1 Confusion matrix

Dealing with unbalanced datasets, the more representative metric for performance evaluation of the model is a balanced accuracy, which is calculated with another two metrics: sensitivity (true positive rate) and specificity (true negative rate).

$$\text{Balanced accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2} \tag{2.2}$$

Sensitivity describes the ability of the model to predict fault class and specificity refers to the prediction of normal operation.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.3}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.4}$$

The second KPI, false alarm rate, is defined in this thesis as the false positive rate, the percentage of times the model has mistakenly flagged the process as faulty (equation 2.5). Together with accuracy, it is used to show one's confidence in the FDD system.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2.5}$$

Detection of a fault is useful, but does little to help in resolving it. Therefore, it is desired that the FDD method not only detects the fault but also shows what kind of the fault it is as well as localizes it in the machine. This indicator is isolation, which describes how well the method determines the specific fault.

Lastly, latency describes the time between the fault occurrence and its detection. For faults of more gradual nature, latency can instead be the level of fault severity, required for it to be detected. Latency is rather fault-specific, and as the time frames of different faults naturally vary, latency will also be different.

Separately, these four indicators can sometimes be very misleading. For example, an overly sensitive FDD method can provide high accuracy and detect even small disturbances, but this can also result in a high false alarm rate, meaning one can rarely trust a positive detection. Optimizing the indicators can be a difficult balancing act, as they oppose each other indirectly. Maximizing one can result in an adverse effect on another. Only when presented together, do they give a good description of the FDD method's capabilities.

2.2 Chillers

Basic principles

A chiller is a device or machine that decreases the temperature by removing heat from the fluid: water circulates in an isolated system, filled with coolant (refrigerant), and by heat exchange process undesirable heat is removed. In addition to application in industrial processes, the chiller is also used for air-conditioning in buildings. Figure 1 shows a simplified schematic of the main components of the water-cooled chiller: evaporator, compressor, condenser, and expansion valve.

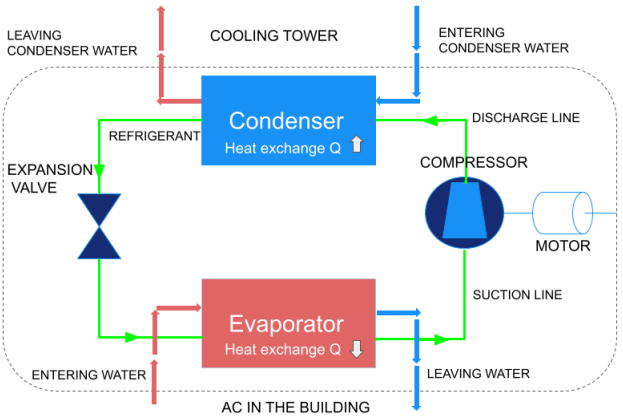

Figure 2.2 Water-cooled chiller diagram

Water from the building enters the evaporator through several tubes and heat exchanges with refrigerant, which initially has a lower temperature. The evaporator shell is filled with refrigerant to a certain level, covering the water tubes. Afterwards, the chilled water leaves the evaporator and goes back to the building, while refrigerant, after absorbing the heat from the water, continues to the compressor in gas form. Different refrigerants have different values of enthalpy for liquid and gas states. The refrigerant is chosen so that it has a boiling point relatively close to the operating water temperature for easier phase transitions. The refrigerant two-phase enthalpy is shown in figure 2.3. In section a-b refrigerant's enthalpy increases in the evaporator, and before entering the compressor, it leaves two-phase form and goes in to superheated gas form.

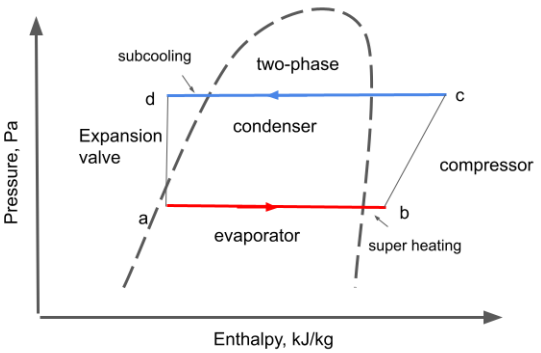

Figure 2.3 Enthalpy two-phase diagram

It is important that refrigerant does not enter the compressor in liquid form as it can damage the compressor. The compressor raises the pressure and temperature of the vaporized refrigerant (b-c), and then it discharges to the condenser. Inside the condenser, the water in tubes absorbs the heat from the refrigerant, which undergoes a phase change into liquid (c-d). Any cooling done after the phase transition into liquid is called subcooling and it generally raises the efficiency of the machine. The released heat then proceeds up to the roof of the building (into an external cooling tower), where it goes out to the surrounding air. The condensed refrigerant goes on to the expansion valve, which divides the high and low-pressure sides of the system. Controlling the refrigerant flow, the expansion valve decreases pressure. The boiling point of the refrigerant is lowered again (d-a), and the refrigerant flows back into the evaporator. The heat transfer process starts again.

Carrier has a variety of chillers, depending on the type of cooling, capacity, application field, refrigerant, and compressor. In this thesis, water-cooled chiller is used for modeling faults.

2.3 Faults

Faults, studied in this thesis, are chosen with respect to their importance and occurrence frequency on different sites and in different conditions, where chillers are installed. The top faults are fouling in the evaporator, fouling in condenser and charge loss (refrigerant leakage). The choice of these faults is also defined by modeling limitations, namely chiller models can simulate only a limited set of faults.

Fouling

Fouling occurs during the cooling process when some unwanted substances (biological, chemical, etc.) accumulate on the surfaces of the heat exchanger tubes. The efficiency of heat flow from an area of higher temperature to an area of lower temperature depends on the thermal properties of components in the thermal circuit. In the evaporator, heat exchange goes from water to refrigerant through a tube surface. These tubes are often made of copper which has high thermal conductivity with a moderate corrosion rate. The surface of the tubes in the evaporator has both external and internal finning enhancements, maximizing mixing and improving heat exchange. The thermal circuit is analogous to an electrical circuit, where the temperature is voltage, heat transfer is current and heat transfer coefficients are resistance. Total thermal resistance is a sum of resistances of refrigerant, copper and water. Fouling factor (allowed coating of the tubes - resistance for fouling) is also considered in the total resistance (equation 2.6):

$$R_{total} = R_{ref} + R_{copper} + R_{foul} + R_{water} \tag{2.6}$$

To increase heat transfer, the thermal resistance must be as low as possible. As fouling adds additional resistance, this results in that refrigerant temperature must decrease to maintain the same heat flow (equation 2.7).

$$Q_{evap} = \frac{T_{water} - T_{ref}}{R} \tag{2.7}$$

Equation below is a linear temperature difference between the refrigerant temperature and the leaving water from evaporator:

$$LTD_{evap} = T_{waterOut} - T_{refrig} \tag{2.8}$$

Leaving water from the evaporator, that goes back to the building, should be constant. Increased LTD_{evap} means that the refrigerant temperature has become lower and this can indicate fouling in the evaporator. LTD might also increase, when there is not enough refrigerant in the system, the water flow rate in the evaporator is decreased or other situations. In this thesis it is assumed that the water flow rate is constant.

Heat transfer (Q) is defined also as a product of heat transfer coefficient (U), area of the heat exchanger (A) and log mean temperature difference (LMTD) (equation

2.9) [Ardsomang et al., 2013].

$$Q_{evap} = U \cdot A \cdot LMTD \tag{2.9}$$

$$LMTD_{evap} = \frac{T_{waterIn} - T_{waterOut}}{\ln \frac{T_{waterIn} - T_{refrig}}{T_{waterOut} - T_{refrig}}} \tag{2.10}$$

Overall heat transfer coefficient (UA) is inversely proportional to the total resistance:

$$UA_{evap} = \frac{1}{R_{total}} \tag{2.11}$$

Control systems hold the heat transfer Q_{evap} constant (in case with constant external parameters) and as fouling increases the thermal resistance, the overall heat transfer coefficient UA decreases, while temperature difference LMTD increases. With higher LMTD, assuming $T_{waterIN}$ and $T_{waterOUT}$ stay constant, refrigerant temperature goes down and therefore also pressure goes down. In this case the compressor starts to work harder to get from low to high pressure.

Similar to the evaporator, heat transfer in the condenser goes from the area of higher temperature to the area of lower temperature, but here the direction of heat transfer is reverse: the coolant is water from the cooling tower and the refrigerant, that flows in the condenser shell, is to be cooled. Heat transfer is then:

$$Q_{cond} = \frac{T_{ref} - T_{water}}{R} \tag{2.12}$$

Equation below is a linear temperature difference between the refrigerant temperature and the leaving water from condenser:

$$LTD_{cond} = T_{refrig} - T_{waterOut} \tag{2.13}$$

And LMTD for condenser:

$$LMTD_{cond} = \frac{T_{waterOut} - T_{waterIn}}{\ln \frac{T_{refrig} - T_{waterIn}}{T_{refrig} - T_{waterOut}}} \tag{2.14}$$

According to the equations above, fouling in the condenser will reduce $LMTD_{cond}$. The refrigerant temperature, and hence also pressure, will go up in the condenser.

Fouling in both evaporator and condenser is often a long process, meaning that dirt can gradually build up for several months before causing a significant capacity reduction. To prevent this, the smallest possible fouling level should be detected in time and this information should serve as a warning, that is supposed to facilitate maintenance services.

Charge loss

The third fault in the thesis, charge loss or refrigerant leakage, results in a reduction of refrigerant level and pressure drop. A condenser has dry and wet tubes, where dry tubes participate in heat transfer and wet tubes - in subcooling. With charge reduction, more dry tubes are included in heat transfer condensation, but UA is not significantly reduced and no pressure change in the condenser will be noticeable while subcooling may become lower. Dry tubes in the evaporator lead to reduced heat transfer coefficients and an increased thermal resistance. In this case the refrigerant temperature is forced to go down, and LMTD increases. Indicators for charge loss are low evaporator pressure or saturation temperature, higher suction superheating, lower condenser subcooling, and higher compressor power.

2.4 Dataset

The data to train and test the FDD algorithm on was simulated by Carrier using one of their dynamic chiller models. To make the data more representative of operational conditions out in the field, the inputs to the model (namely the entering water temperatures in the evaporator and the condenser) were fetched from a real chiller, while the setpoint (the leaving evaporator water temperature) was set to be constant. The dataset consists of 35 time periods of operation during different days, each such period roughly 11 hours long. The vast majority of those periods are from the month of July, with only a few being from August and September. The entering water temperatures of each period were then used as inputs in the chiller model to generate a total of 54 samples per period. A sample in this case is a specific preset of parameters that are used to simulate the faults at varying severities. Their structure is as follows: samples 0, 10 and 20 are identical and are generated using normal parameters, simulating the behavior of a chiller when no faults are present. These three samples are identical and so only sample 0 was used. Samples 1-9 had the thermal resistance between the refrigerant and the water in the evaporator increased gradually, simulating fouling in the evaporator at rising severity levels. Similarly, samples 11-19 had increased thermal resistance in the condenser to simulate fouling there. Charge leak was simulated by reducing the volume of refrigerant in the system for samples 21-29. These samples have no more than a single fault active and so are referred to as OAT (One At a Time) fault samples. The last 24 samples have multiple faults present simultaneously at varying levels. In the simultaneous faults, each of the different fault parameters was held at either the normal or the least severe faulty values, while the other two faults were varied. This was done to check if the FDD algorithm would confuse two simultaneous faults for another. The faults severities can be seen in tables 2.1 and 2.2. Each sample contains a total of 29 signals. These include signals like the incoming water temperatures that were used as input to the models, diagnostic signals like LTD and LMTD and others. As the data comes from a simulation and not a physical chiller, it allows for monitoring of system variables that are not currently measured out in the field. An example of such a signal is subcooling. As this data was generated in a simulation the sampling time isn't constant with most of the samples being concentrated at the start of the simulation i.e. the transient part.

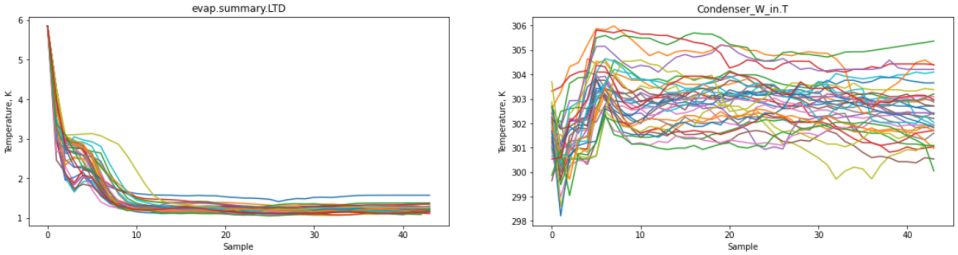

Figure 2.4 A diagnostic signal from simulation on the left (LTD in evaporator) and an input signal from field data on the right (incoming condenser water temperature) under normal operation parameters (sample 0). The data point frequency is reduced to 1 per 15 minutes for all 35 time periods for non-faulty operation, each represented by a different line in the figure.

Table 2.1 Normalized fault severity levels for OAT faults. Evap_foul and Cond_foul are both nominal heat transfer resistances, while Charge_loss is the nominal volume of refrigerant in the system.

Sample	Evap_foul	Cond_foul	Charge_loss
0	1.00	1.00	1.000
1	1.56	1.00	1.000
2	2.11	1.00	1.000
3	2.67	1.00	1.000
4	3.22	1.00	1.000
5	3.78	1.00	1.000
6	4.33	1.00	1.000
7	4.89	1.00	1.000
8	5.44	1.00	1.000
9	6.00	1.00	1.000
10	1.00	1.00	1.000
11	1.00	1.27	1.000
12	1.00	1.53	1.000
13	1.00	1.80	1.000
14	1.00	2.07	1.000
15	1.00	2.33	1.000
16	1.00	2.60	1.000
17	1.00	2.87	1.000
18	1.00	3.13	1.000
19	1.00	3.40	1.000
20	1.00	1.00	1.000
21	1.00	1.00	0.993
22	1.00	1.00	0.985
23	1.00	1.00	0.978
24	1.00	1.00	0.971
25	1.00	1.00	0.963
26	1.00	1.00	0.956
27	1.00	1.00	0.949
28	1.00	1.00	0.941
29	1.00	1.00	0.934

Table 2.2 Normalized fault severity levels for simultaneous faults. Evap_foul and Cond_foul are both nominal heat transfer resistances, while Charge_loss is the nominal volume of refrigerant in the system.

Sample	Evap_foul	Cond_foul	Charge_loss
30	1.00	1.27	0.941
31	1.00	2.60	0.985
32	1.00	1.27	0.985
33	1.00	2.60	0.941
34	1.56	1.27	0.941
35	1.56	2.60	0.985
36	1.56	1.27	0.985
37	1.56	2.60	0.941
38	1.56	1.00	0.941
39	4.33	1.00	0.985
40	1.56	1.00	0.985
41	4.33	1.00	0.941
42	1.56	1.27	0.941
43	4.33	1.27	0.985
44	1.56	1.27	0.985
45	4.33	1.27	0.941
46	1.56	2.60	1.000
47	4.33	1.27	1.000
48	1.56	1.27	1.000
49	4.33	2.60	1.000
50	1.56	2.60	0.985
51	4.33	1.27	0.985
52	1.56	1.27	0.985
53	4.33	2.60	0.985

2.5 Machine learning algorithms

Machine learning algorithms learn from training over data in order to recognize certain patterns and predict targets. The problem formulated in the thesis is a supervised classification task, where the discrete class labels must be predicted. The data, that is used to train a machine learning model, is called training data and consists of some amount of input-output data samples. Inputs are variables that are somehow related to outputs, while outputs in the classification task are discrete values. In this thesis, outputs are labels for a respective fault or normal operation class and these labels are chosen to be integer values. There are several types of supervised machine learning algorithms that can solve classification tasks, and, having the focus on the implementation of a simple model with high interpretability, three methods are chosen: decision tree, logistic regression and support vector machine. Moreover, building algorithms based on these models is not so complex and the amount of hyperparameters is not so big. Hyperparameters are parameters that control the learning process and they are set before the process starts, usually as arguments in a constructor of a model. In this thesis, these three methods are created by using implementations from respective Scikit learn libraries [Pedregosa et al., 2011]. All machine learning methods have their own set of hyperparameters, that should be tuned in order to get a high generalization level of the model and avoid overfitting, which might occur when the model is trained too much on details of data. There are two techniques, called L1 and L2 regularization, that help to prevent overfitting and will be discussed later.

Loss function and regularization

For estimation of how well data is modeled, a loss function is applied to optimize the learning process of a model and the goal of optimization is to minimize this loss. For binary classification problems the loss function for logistic regression is binary cross entropy, that compares predicted values with actual outputs:

$$\mathrm{Loss_i} = -[\mathrm{y_i} \log \hat{\mathrm{y}}_\mathrm{i} + (1 - \mathrm{y_i}) \log (1 - \hat{\mathrm{y}}_\mathrm{i})], \tag{2.15}$$

where y_i and $\hat{y}_i$ are true respectively predicted outputs. N is amount of data instances. Cost function is an average loss over the entire training data:

$$\mathrm{Cost} = -\frac{1}{\mathrm{N}} \sum_{\mathrm{i}}^{\mathrm{N}} [\mathrm{y_i} \log \hat{\mathrm{y}}_\mathrm{i} + (1 - \mathrm{y_i}) \log (1 - \hat{\mathrm{y}}_\mathrm{i})], \tag{2.16}$$

The aim of this optimization task is to find an optimal set of model parameters that minimizes the value of the cost function. The reduction of cost means the reduction of error and thereby increased model performance. Binary cross-entropy equals negative log-likelihood, which means that minimizing the cost based on cross-entropy is the same as maximizing the likelihood. Regularization techniques add a penalty for increased complexity of the model, which helps to overcome overfitting of the

model. There are two different regularization terms - L1 and L2, and they are added to the cost function. L1 regularization adds absolute values of weight multiplied by regularization rate λ, which influences regularization strength. A high value of λ increases regularization and regularization hyperparameter C, used from Scikit learn implementation, is inversely proportional to λ.

$$L1 = \lambda \sum_{i}^{N} |\omega_i| \tag{2.17}$$

With L1 regularized cost function less important features get zero weights. L2 regularization is a sum of squared weights multiplied by regularization rate, it penalizes insignificant features but instead of setting these weights to zero as in L1 regularization, it just constricts the weights.

$$L2 = \lambda \sum_{i}^{N} |\omega_i|^2 \tag{2.18}$$

These two techniques are used to overcome overfitting and find the best trade-off between bias and variance, but their applications differ, depending on the goal of regularization. L1 regularization can be used for feature selection while L2 is better for learning from more complex data patterns as it takes into consideration all features [Mohri et al., 2018].

Logistic regression

Logistic regression (LR) is a parametric method that predicts probabilities of a target using independent features and the output, in that case, must be binary. Using the cross-entropy loss and optimization, defined by the choice of the solver (discussed later), trains a linear relationship from the given dataset:

$$y = \mathbf{w}^T \mathbf{x} + b, \tag{2.19}$$

where w and x are weight resp. feature vectors and b is intercept (bias). After this a sigmoid function (equation 2.20) maps outputs from the model between 0 and 1 as seen in figure 2.5 and returns the probabilities of classes.

$$S(y) = \frac{1}{1+e^{-y}} = \frac{1}{1+e^{-(\mathbf{w}^T \mathbf{x}+b)}} \tag{2.20}$$

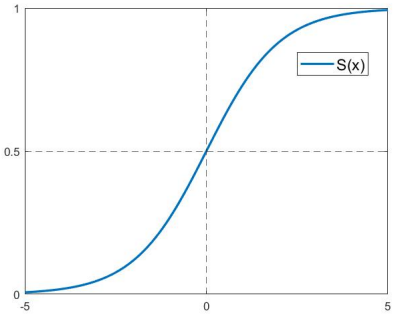

Figure 2.5 Sigmoid function

The main hyperparameters for logistic regression (discussed below) are:

1. Solver
2. Regularization
3. Regularization parameter
4. Number of iterations
5. Class weight

There are several solvers in the Scikit learn implementation to choose from: newton's method, limited-memory broyden–fletcher–goldfarb–shanno algorithm (lbgs), liblinear, stochastic average gradient with only L2 regularization (sag) and stochastic average gradient with both regularization techniques (saga). They all have different application fields, for example, liblinear is good for small datasets and sag or saga works faster for larger ones. Logistic regression is an iterative process, which means that an optimal number of iterations must be set in order to achieve convergence of the learning process.

The main limitation of logistic regression is that it can construct only linear decision boundaries and linearly separable data is very rare in real-world implementations. If data is linear and two features are enough to separate it, the boundary can be easily visualized in a 2D plot. Logistic regression is also prone to outliers and unbalanced data, therefore pre-processing of data might be essential.

Support vector machine

Support vector machine (SVM) is another supervised learning algorithm that can solve classification problems and like the previous one, it also builds a decision boundary between classes. SVM is a maximum margin method and it tries to find a boundary with the maximum margin to the closest data instances. The classification

of two classes is shown in figure 2.6, where the decision boundary - the separating hyperplane (a line for two dimensions), divides data instances and assigns them to a respective class. Data instances, closest to the separating hyperplane, define support vectors.

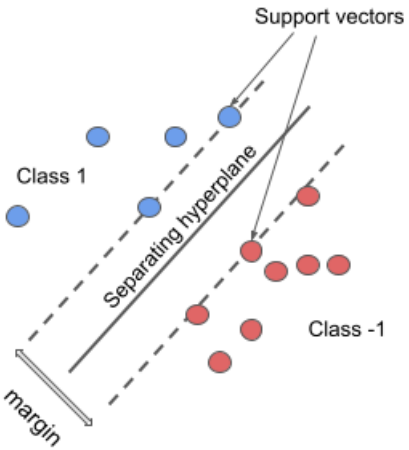

Figure 2.6 Linear data separation with SVM. The separating hyperplane in this case is a line that divides two classes with the largest margin. Margin is the maximum distance between hyperplanes for support vectors (dashed lines). Data samples lying on these hyperplanes are called support vectors and they define the position of these hyperplanes.

As seen in the figure, there are no data points (in the separable case) between the separating hyperplanes, which means that the basic support vector machine does not allow misclassification, and the margin is called the hard margin. The object of optimization for SVM is to find the best and largest margin between classes. The best support vectors, the data instances closest to the boundary hyperplane, are defined by the largest margin to the boundary [Schölkopf, 2002]. With maximized margin, the model will generalize better on unseen data instances. However, the hard margin works well for linearly separable data, but in real-world applications data is often non-linear. In this case, a soft margin can be used as it tolerates some errors.

Besides that SVM finds the optimal decision boundary, other advantage of the method is a technique, called kernel trick, that can transform data into a higher dimension and also allows to solve non-linear problems. Kernel is a function, that can be linear, polynomial, sigmoid, Gaussian and others. This technique gives opportunity to fit both linear and non-linear decisions, making SVM rather powerful and a flexible tool for machine learning problem. SVMs can be computationally faster than logistic regression, depending on the choice of the kernel function, but it also

may be less efficient if data is noisy and classes may overlap each other. Moreover, being not a scale-invariant method, the data must be scaled before training the model.

The main hyperparameters for SVM are:

1. Kernel
2. Regularization parameter
3. Class weight

Decision tree

A decision tree is a non-parametric supervised method, that uses one feature for data separation into classes in each node and looks like a flow chart (see figure 2.7).

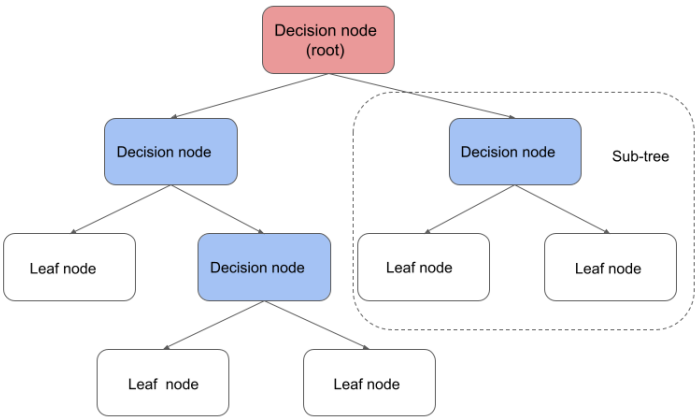

Figure 2.7 Schematic of decision tree

Using a branching approach, the decision tree takes a feature in the root to split data points into subsets of nodes with some measurement of the quality of the resulting split of the data (gini or entropy index). This procedure is repeated until the stopping criteria is reached. Stopping criteria can be, for example, the maximum depth of the tree or other hyperparameters, that can be tuned for the tree. The last nodes in the tree, leaves, represent class labels.

The main hyperparameters for the decision tree (discussed below) are:

1. Criteria: gini/entropy index
2. Maximum depth of the tree

3. Minimum samples for split
4. Minimum samples for a leaf node
5. Maximum features
6. Class weight

Two criteria, either gini or entropy impurity, are used in the decision tree to measure the quality of a split and by these parameters, the optimal split of features is found. Gini impurity refers to how many instances are mislabeled.

$$\text{Gini index} = 1 - \sum_{j} |p_j|^2, \tag{2.21}$$

where p_j is a probability of class j. Gini impurity has range between 0 and 0.5 for a binary case. Zero gini index means that the node is pure, namely all instances in the node are correctly classified to one class, while index 0.5 tells that both classes have the same probability. Entropy measures a level of disorder and like gini index, the split is perfect when entropy index is zero. As entropy lies between 0 and 1, the maximum entropy index is 1.

$$\text{Entropy index} = -\sum_{j} p_j \log(p_j) \tag{2.22}$$

Both these metrics calculate an information gain of features and are used in the decision tree to find the best splitter for each node.

The main advantage of the decision tree is that it is easy to visualize and interpret the model as it is closely related to the human way of decision making. Moreover, data does not need to be scaled, but as the decision tree is prone to overfitting, hyperparameter tuning has a crucial role in good model performance [Lindholm et al., 2022]. An additional disadvantage is that small changes in data can lead to another structure of the tree, which makes this method more or less unstable. Another machine learning method, called random forest, can usually enhance the performance as it uses ensembles of decision trees and makes predictions by majority vote from the trees. At the same time, it costs the ability to interpret the decision-making process as one can in the single decision tree case.

3

Methodology

3.1 Initial work

This project was very iterative in nature, both the dataset and the methods used to analyze it have changed many times throughout the project. This means that the methodology described in the following sections of this chapter as well as the results in the next chapter do not accurately portray the total amount of work done as they only describe the final iteration. The authors have therefore decided to include this section where the summary of the work process can be described chronologically albeit with less detail.
The initial plan on how the data was to be analyzed was done before any data was generated. The dataset was to be divided into several parts and the signals in these parts were to be condensed down to several different features. The features were decided to be statistical 1-4th order cumulants as well as the maximum and the minimum values of each signal within the boundaries of that part. The method chosen first was the decision tree as it was very interpretable, allowing us to directly see what the labeling decisions were based on.
The first iteration of data was generated on a simpler, steady-state chiller model. Because of the small size of the dataset, it became apparent that some tricks would have to be adapted so that enough data samples could be generated to train the machine learning algorithm. The sliding window method, which is described in greater detail in section 3.2, was implemented as well as the size of these windows was made small. The small window size and the relatively slow dynamics in the signals resulted in many signals being constant within the window, which caused stability issues in the code as the higher-order cumulants become undefined. Additionally, since this was a steady-state model, the faults did not affect the signal in any dynamic way and only resulted in a static offset in several signals. This made cumulants 3 and 4 both problematic and practically obsolete which ultimately led to their removal. It was also noted that the initial batch of signals provided was not well suited for the detection of the selected faults and that a bigger and/or more carefully selected set of signals needed to be produced.
Following that, the dataset has gone through many iterations. Both the model and

the input data have changed multiple times until the final version was settled. Several of these iterations had problems with either the model or the parameters which are difficult to spot directly, as it requires a good understanding of chiller dynamics. The list of signals changed as well, many signals that were heavily correlated were removed. A large signal set was generated and then pruned with the help of feature importance features within the different ML models, and with help from Carrier physics informed signals were introduced. An attempt was made to analyze the transient data during chiller startup by removing the steady-state data and introducing correlation between each signal within the window as features in form of Pearson correlation. However, after being unable to produce promising results with this strategy, it was abandoned in favor of the previous one. The dataset and the model were finalized around the halfway point of the project and the lessons learned during the iterations as well as the methods described below could finally be applied.

3.2 Data pre-processing

The data had to be pre-processed before it could be used to train the FDD algorithms. This included the following steps: data inspection, interpolation, creating more data samples from given data, feature engineering, and normalization of data.

First of all, some of the samples in different periods had a single data point per signal, presumably because the simulation crashed or did not converge to a solution. This likely happened because the parameter changes that were used to simulate the faults pushed the system outside of its normal operating range as all "broken" samples had a fault in them and the vast majority of them had multiple faults simultaneously.

Second, the data had to be interpolated to achieve a static data point frequency. It was then also reduced to a frequency of 1 per 15 minutes as this is the frequency currently used in the chillers out in the field. The resulting data now consists of 44 data points for each of the 29 signals in every sample.

The data was then further divided using a sliding window method, which is shown in figure 3.1 and the features of each window (data sample) were computed. Figure 3.2 illustrates how from 29 signals a training dataset for one fault was created: each signal, using the sliding window method, produces 21 data samples, that are then each condensed to 4 features (described in 3.3) and finally, the training dataset is formed for all samples.

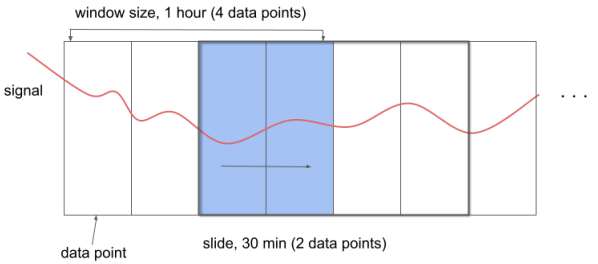

Figure 3.1 Illustration of the sliding window method, that produces more data samples for training data, but they are no longer independent of each other. A window consists of 4 data points from the signal. It then slides 2 data points, creating another window, which partly overlaps with the previous one. As each of 35 periods has signals for 11 hours, totally there are 44 data points and by the sliding window method 21 data samples are produced from one signal.

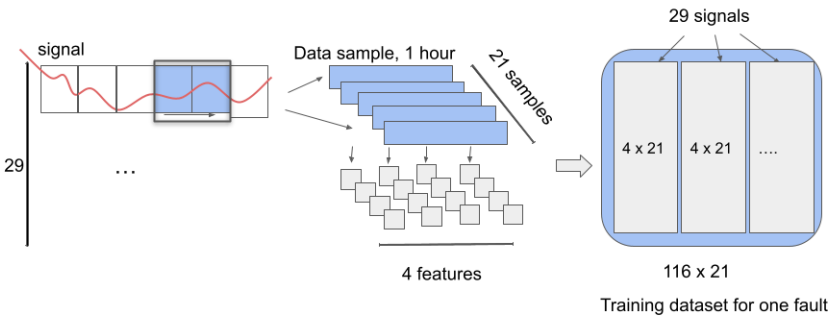

Figure 3.2 Illustration of training data set generation for one fault. There are 29 signals in total, each signal produces 21 data samples from sliding window method and then 4 features for each data sample are computed. Training dataset for one fault gets dimension of 116 features (for all signals) and 21 data samples.

Machine learning models estimate weights for each data point, and as the distribution of data points can be different for each feature, it is important to scale data for parametric methods, logistic regression and support vector machine. Normalization, one of the scaling methods, is done for both training and test data, where each feature value is calculated by formula 3.1. The scale is estimated from training data and then the same scale is applied to test data.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

3.3 Feature Selection

In this project, two different approaches are used to identify which signals and features actually contribute mostly to the results and which are redundant. The first approach uses physics-based assumptions about what signals might be useful for prediction. These were made by professionals working at Carrier with a good understanding of the physics of chillers. The second one is a statistical approach using tools from Scikit learn.

Features are created by condensing each signal to 4 features: first and second-order cumulants (mean value and standard deviation), minimum and maximum values. Following physics-based assumptions about the signals (see section 2.3), potentially good indicators of each fault are:

- Fouling in evaporator: high LMTD, LTD for evaporator
- Fouling in condenser: high LMTD, LTD for condenser
- Charge loss: low condenser subcooling, high compressor power, low evaporator pressure

All provided signals from the simulation are included in feature construction with the aim to reduce them to a minimal amount with respect to the highest KPIs. Features were selected with the help of feature importance for each machine learning method. The decision tree method has feature importance estimation with respect to mean impurity decrease and also mean accuracy decrease. Mean impurity decrease is defined as the total decrease in node impurity: the value that tells how well the feature separates data points into classes. If nodes get less impure data in classes, the feature, used for split, is considered important. Mean accuracy decrease, also called permutation importance, is a technique when values from one feature are shuffled and a change in model performance is observed. If the accuracy drops, the features are considered to be important. The major contributors to LR and SVM training are found by sorting the highest absolute values of the weights assigned to the features. In binary classification, one class gets positive coefficients and the second one - negative coefficients, therefore absolute values are used for ranging the most important feature for both classes.

After statistical selection of the features, they are compared to the physics-based features, that were previously assumed to be the best ones for fault identification. Linear separability is rare, but a desirable property of data as it indicates that the problem can be solved with simple models. With linear separable data, classes can be separated by a line for two dimensions, or a hyperplane for higher dimensions. For SVM and LR it might be useful to have 2-3 features in order to provide good visualization for separation in 2D or 3D plots. With more than 3 features, it is not possible to plot the hyperplanes in an interpretive way.

3.4 Machine learning methodology

Machine learning implementation consists of the following steps:

1. creating a model
2. training the model on the data and validation of the model
3. hyperparameter tuning
4. testing trained model on test data
5. evaluation of the model performance

This process is iterative, meaning that depending on the result from one step, it might be needed to return to the previous steps and do some changes.

Binary classifiers

Data that must be classified into four classes: three faults and normal operation, infers a multiclassification task, but there are some techniques for transforming the task to a binary one. One-to-rest transformation is applied to the problem and to implement FDD algorithm three separate binary classifiers, one for each fault, are created. These have samples of a respective class labeled 1, and others 0. For example, the condenser classifier has label 1 for the fault, which is to be detected by this classifier fouling in the condenser, but label 0 is set for fouling in the evaporator, charge loss, and also normal operation (figure 3.3). The same principle works for the other two classifiers.

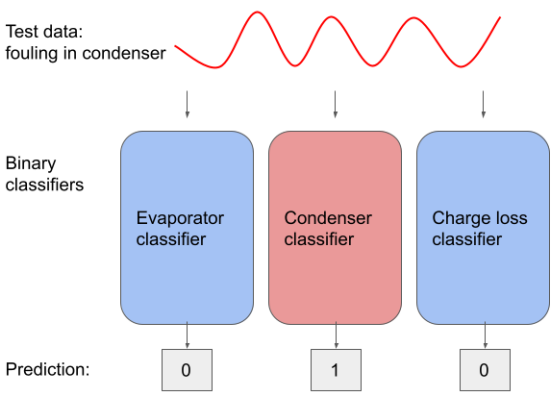

Figure 3.3 Test signal for fouling in condenser in three classifiers: evaporator classifier predicts as no fault, label 0; condenser classifier - fault is present, label 1; and charge loss classifier - no fault, label 0.

The choice of using binary classifiers is motivated by the requirement of fault isolation (as one of KPIs) and it is assumed to be an easier way for performance evaluation: performance scores for each fault can be evaluated separately. Another reason for binary transformation is that one of the project's goals is to use simple machine learning models with only a few hyperparameters and features for each fault.

Training and validation

After data is pre-processed and labeled, each classifier is trained on the data with the same features but different labeling and is validated by the cross-validation technique. The training dataset is divided into several equal parts of data, one part is reserved for validation, while the model is trained on the other parts of the data. The process is repeated until all parts are involved in validation. Scoring metrics from each validation part averaged. Validation scores serve as indicators of how well the model generalizes to an unknown dataset and may show a problem with overfitting or bias. As datasets are composed from 35 days into data samples of size 1 hour (using the sliding window method), one case has 21 data samples for one fault, one day, and one severity level. Training of the model is done either on 34 days (one day is for testing) or 1-5 days. Portioning of training data is set to 10 parts in cross-validation with shuffling and the model is validated on randomly picked hours from different days. Shuffling gives more representative validation scores as the data is initially sorted by classes. There are 3 different sets for training data composition, that are used in the thesis:

1. Complete data (including transient part), all signals, all features
2. Steady-state data, all signals, mean values
3. Steady-state data, the most important signals, mean values

Each of these sets has also three different combinations of running the training and testing of the data, called cases:

- Case 1: train on 34 days and test on 1 day
- Case 2: train on 5 days and test on 30 days
- Case 3: train on 1 day and test on 34 days

Having 3 classifiers, 3 machine learning models, 3 cases for training and testing combination, and 3 training data compositions, we got a total of 81 iterations of machine learning process.

Hyperparameters

Regarding hyperparameters for the machine learning models, the first approach is to run the models with default hyperparameters from Scikit Learn libraries for each model. Depending on the performance scores from cross-validation, some essential tuning may have to be done. Some hyperparameters are from the beginning already known to stay unchanged: class weight is set to be balanced as one class is always underrepresented in each classifier: one fault class is just one-quarter of all data samples in the classifier. For the support vector machine model, a linear kernel is chosen as a start point. Regularization is set to be L2 with default regularization parameter C, set to 1, for both logistic regression and support vector machine. The number of iterations for logistic regression is 200, but it possibly should be increased if the model does not converge.

Model performance

Evaluation of model performance is done with 4 KPIs: accuracy, false alarm rate, isolation, and latency. In this case, latency is defined as the severity of the fault before it can be accurately detected. This was decided to be reasonable as all of the mentioned faults usually grow very slowly and the faulty samples contain the fault from the start-up, i.e. there is no time event where the fault is introduced into the system. Because the FDD algorithm is being implemented as three separate binary classifiers, the isolation metric becomes the accuracy ratings of the classifiers. Due to the unbalanced test data, it is chosen to use a balanced accuracy, which considers the number of data samples for each class. A false alarm rate is defined as a false positive rate, the percentage of positive cases. A confusion matrix is used as one of the tools that allow good visualization of performance measurements (TP, TN, FP, FN) for predicted output.

Target for KPIs is:

- Accuracy - 95 %
- False alarm rate - 5 %
- Latency - one of 9 fault severity levels
- Isolation - 95% accuracy for each classifier

4

Results

4.1 Machine learning models

Some of the hyper-parameters (discussed in chapter 2.5) like class weight, kernel, and criteria were set from the beginning of training process and stayed unchanged, while others were obtained after searching for the best validation scores. Parameter tuning was considered to be optimal after validation scores showed 97-98%. Both default and tuned hyper-parameters for the models are presented below:

1. Decision tree:
    - Criteria: gini
    - Maximum depth of the tree: 40
    - Minimum samples for split: 2
    - Minimum samples for a leaf node: 2
    - Maximum features: sqrt
    - Class weight: balanced
2. Logistic regression:
    - Solver: saga
    - Regularization: L2
    - Regularization parameter: C=1
    - Number of iterations: 200
    - Class weight: balanced
3. Support vector machine:
    - Kernel: linear
    - Regularization parameter: C=1 (regularization: L2)
    - Class weight: balanced

4.2 Performance scores: complete dataset

At first, the complete dataset was used, when all 29 signals together with 4 features resulted in 116 features (mentioned in 3.1). The training and test data were divided into 3 cases according to the table 4.1. Training data included training for all 9 severity fault levels of each fault, while the test data had only the lowest fault severity.

For case 1 training dataset got more than 25 thousand data samples from 34 days, the biggest test data (case 3) was also composed from 34 days, but as it included only one fault level the size of the data was about 2800 data samples.

Table 4.1 Complete dataset. Number of data samples for 3 cases for training and test data.

	Training data		Test data	
	Days	Data samples	Days	Data samples
Case 1	34	25 704	1	84
Case 2	5	3 780	30	2 520
Case 3	1	758	34	2 856

The following tables 4.2 - 4.4 are the results for One-at-time fault (OAT) detection, where the average KPIs for three cases are summarized. As it is seen from the tables, the SVM model manages to generalize better than the other two, especially when it is enough training data (5-34 days). Scores for LR lie rather close to SVM scores, but the decision tree performed significantly worse with a reduced amount of training data.

Table 4.2 Average performance scores for complete data, case 1: trained on 34 days, tested on each of 35 days, 116 features.

Classifier	**Models**			**The most important features**
	SVM	**LR**	**Tree**	
Evaporator				
Accuracy,%	100	100	98.5	evap.LTD,T suction,
False alarm rate,%	0	0	0.19	T exp.valve
Condenser				
Accuracy (%)	100	100	92	
False alarm rate,%	0	0	3	cond.LTD,subcooling
Charge loss				
Accuracy,%	98	90	83	
False alarm rate,%	1.9	2.1	9	cond.rellevel, subcooling

Table 4.3 Average performance scores for complete data, case 2: trained on randomly chosen 5 days, tested on 30 days, 116 features.

Classifier	Models			The most important features
	SVM	LR	Tree	
Evaporator				
Accuracy,%	99.2	98.9	95	evap.LTD,T suction,
False alarm rate,%	0.8	1.4	5	T exp.valve
Condenser				
Accuracy,%	98.9	98.7	86.8	
False alarm rate,%	0.4	0.6	10	cond.LTD, subcooling
Charge loss				
Accuracy,%	97.2	88.4	78.7	cond.rellevel, subcooling
False alarm rate,%	2	2.3	10	

Table 4.4 Average performance scores for complete data, case 3: trained on 1 day, tested on 34 days, 116 features

Classifier	Models			The most important features
	SVM	LR	Tree	
Evaporator				
Accuracy,%	96.4	91.7	88.5	LTD,T suction,
False alarm rate,%	2.6	3.4	10	T exp.valve
Condenser				
Accuracy,%	96.8	81.9	69.2	
False alarm rate,%	2	4	38	cond.LTD, subcooling
Charge loss				
Accuracy,%	79.1	69.3	63.2	cond.rellevel, subcooling
False alarm rate,%	16	13	19	

Table 4.4 shows average performance scores for the models, where each of them was trained on 1 day and tested on 34 days. This case was repeated 35 times. Common for all three models is that charge loss is the most challenging part for fault detection and with only one day for training (case 1) even SVM fails to achieve the KPIs: accuracy is 80% (KPI is minimum 95%) and false alarm rate 16% (KPI is max 5%).

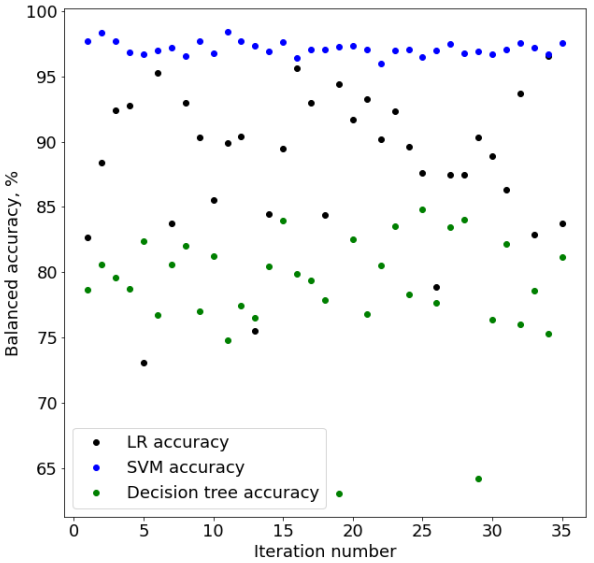

Figure 4.1 Accuracy scores for charge loss for case 2: trained on 5 randomly chosen days and tested on the rest of 30 days (see table 4.3). This test was repeated in total 35 times (iteration number on the axis)

Figure 4.1 show accuracy scores for charge loss for three models. This is case 2 when they are trained on randomly chosen 5 days and tested on 30 days. This process is repeated 35 times to get a reliable average value for accuracy. It is seen clearly, that SVM (blue dots on the plot) lie all the time between 95-100%, while the other two models have much more variations between 65-95%.

4.3 Performance scores: steady state data

To improve the performance of the algorithm for this specific dataset, the transient part of the data (approximately 3 hours from the beginning of the 11 hour period) was removed. Due to the stationary nature of the steady state operation, only one feature per signal, mean value, was used and the second-order cumulant, as well as the min and max values, were removed. Totally 29 features (the same amount as signals) are used for training and testing. Table 4.5 shows the dimensions of the reduced dataset.

Table 4.5 Steady state data. Number of data samples for 3 cases for training and test data.

	Training data		Test data	
	Days	Data samples	Days	Data samples
Case 1	34	18 360	1	60
Case 2	5	2 700	30	1 800
Case 3	1	540	34	2 040

Tables 4.6 - 4.8 present the average KPI values for 3 cases.

Table 4.6 Average performance scores for steady state data, case 1: trained on 34 days, tested on each of 35 days, 29 features.

Classifier	**Models**			**The most important features**
	SVM	**LR**	**Tree**	
Evaporator				
Accuracy (%)	100	100	98.5	evap.LTD,T suction,
False alarm rate,%	0	0	0.19	T exp.valve
Condenser				
Accuracy,%	100	100	98.1	cond.LTD, subcooling
False alarm rate,%	0	0	0.3	cond.W_{in}, condensing T
Charge loss				
Accuracy (%)	100	100	95.9	cond.rellevel,subcooling,
False alarm rate,%	0	0	1.3	PID exv,exv out T

Superior results are seen for case 1 for all three models: SVM and LR get 100% for all faults and the decision tree achieves scores above the baseline for KPIs. This is consistent with what has been assumed that steady-state data should improve performance.

Table 4.7 Average performance scores for steady state data, case 2: trained on randomly chosen 5 days, tested on 30 days, 29 features.

Classifier	Models			The most important features
	SVM	LR	Tree	
Evaporator				
Accuracy,%	100	100	99.7	evap.LTD,T suction,
False alarm rate,%	0	0	0.19	T exp.valve
Condenser				
Accuracy,%	100	100	91	cond.LTD, subcooling,
False alarm rate,%	0	0	7	cond.W_{in}, condensing T
Charge loss				
Accuracy,%	99.9	99.99	86.2	cond.rellevel, subcooling
False alarm rate,%	0.004	0.05	5	PID exv, exv out T

Table 4.8 Average performance scores for steady state data, case 3: trained on 1 day, tested on 34 days, 29 features.

Classifier	Models			The most important features
	SVM	LR	Tree	
Evaporator				
Accuracy,%	99.2	98.7	87.7	evap.LTD,T suction,
False alarm rate,%	1.7	2.1	12.5	T exp.valve
Condenser				
Accuracy,%	99.4	96.9	70.6	cond.LTD,
False alarm rate,%	1.2	1.3	38	cond.W_{in}, condensing T
Charge loss				
Accuracy,%	99.8	98	67.9	cond.rellevel, subcooling
False alarm rate,%	0.3	0.05	21	PID exv, exv out T

Almost similar scores are obtained in cases 2 and 3 for SVM and LR, while the decision tree is the weakest model and for charge loss accuracy decreased to 68%. This is also illustrated in figure 4.2, where accuracy for three models is plotted for each training day (case 3). In comparison with the result for a 1-day training with complete data and 116 features, charge loss detection is quite improved for both SVM and LR.

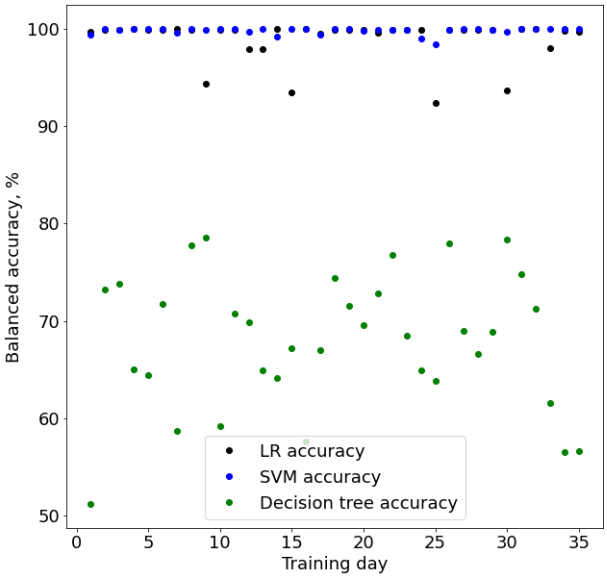

Figure 4.2 Accuracy scores for charge loss for case 3: trained on 1 day and tested on other 34 days. SVM confirms again that it the best model of these three: for all training days accuracy lies close to 100%.

4.4 Signal reduction

As it was mentioned in section 3.3 for feature selection, the amount of signals reduced with feature importance analysis for respective machine learning model. All three models got the same important features. Minimum sets of signals, essential for linear separation of the faults with respect to outlined KPIs are presented in table 4.9 (for case 2 and 3). Fouling in evaporator and condenser have got two different sets of the most significant features that achieved linear separation for these faults. Charge loss fault was found to need at least three features to be separated.

Table 4.9 The most significant signals for detecting malfunction due to fouling in evaporator, fouling in condenser and charge loss, needed for linear separation.

	Signals	
Fouling in evaporator	1) evaporator LTD 2) condenser LTD	1) condenser LTD 2) expansion valve
Fouling in condenser	1) evaporator LTD 2) subcooling	1) evaporator LTD 2) expansion valve
Charge loss	1) condenser LTD 2) subcooling 3) expansion valve	

According to the previous results for the case with 29 signals, case 1 was excluded as it was proven that 5 days of training (case 2) provides similar high scores as with 34-days training (case 1). After feature analysis, the following 4 features are selected: LTD for the evaporator, LTD for the condenser, expansion valve (output from the controller), and subcooling. Average performance scores for case 2 and case 3 with selected signals are summarized in two tables below.

Table 4.10 Average performance scores for steady-state data, case 2: trained on 5 days, tested on 30 days, the 4 most important features.

Classifier	**Model**		
	SVM	**LR**	**Tree**
Evaporator			
Accuracy, %	100	100	92
False alarm rate. %	0	0	4
Condenser			
Accuracy, %	100	99.9	90.2
False alarm rate, %	0	0	6
Charge loss			
Accuracy, %	99.8	98.2	86.4
False alarm rate, %	0	0	8.4

Table 4.11 Average performance scores for steady-state data, case 3: trained on 1 day, tested on 34 days, the 4 most important features.

Classifier	Model		
	SVM	**LR**	**Tree**
Evaporator			
Accuracy, %	100	86	78
False alarm rate. %	0	0	2
Condenser			
Accuracy, %	99.5	80.2	70.9
False alarm rate, %	0.6	9	37
Charge loss			
Accuracy, %	95.5	76	69
False alarm rate, %	0	0.08	27

From table 4.11 it is seen that scores for SVM are still good with respect to KPI baseline, while logistic regression and decision tree perform significantly worse. Some parameter tuning was done for both of them and the biggest change in performance is noticed for logistic regression when L1 regularization is applied instead of L2 regularization. Table 4.12 shows scores for LR with these two regularizations. With L1 regularization the scores are improved, including charge loss accuracy that increased from 76% to 95% (see figure 4.3), achieving the goal for this KPI. A significant reduction of false alarm rate is noticed for condenser fouling as it was reduced from 9% to 1%, again meeting the requirement.

Table 4.12 Average performance scores for Logistic regression with L1 and L2 regularization. Steady-state data, case 3: trained on 1 day, tested on 34 days, the 4 most important features. L1 regularization increased the performance scores for logistic regression, that the outlined KPIs are now met. (>95% accuracy and <5% false alarm rate)

Classifier	Model	
	LR (L2)	**LR (L1)**
Evaporator		
Accuracy, %	86	99.5
False alarm rate, %	0	0
Condenser		
Accuracy, %	80.2	98.5
False alarm rate, %	9	1
Charge loss		
Accuracy, %	76	95
False alarm rate, %	0.08	0

The accuracy scores for charge loss for the SVM, LR and decision tree models are illustrated in figure 4.3. On the left LR has L2-regularization and on the right - L1-regularization. It is seen on the both plots that for some days (between 22-25), used for training in case 3, the performance was significantly worse for all three models. Comparing the input signals (entering water temperature in the evaporator and the condenser) with other days, high values of the temperatures were noticed and the signals of these days had much more variations. Therefore, to have some of these particular days for training was not enough to generalize the models well.

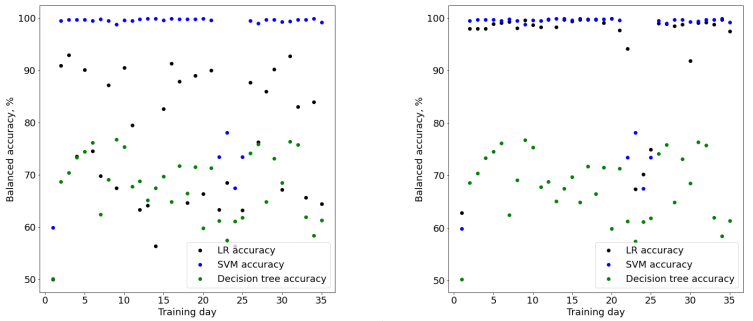

Figure 4.3 (a) Accuracy scores for charge loss, the 4 most important features (LR has L2-regularization) (b) Accuracy scores for charge loss, the 4 most important features (LR has L1-regularization)

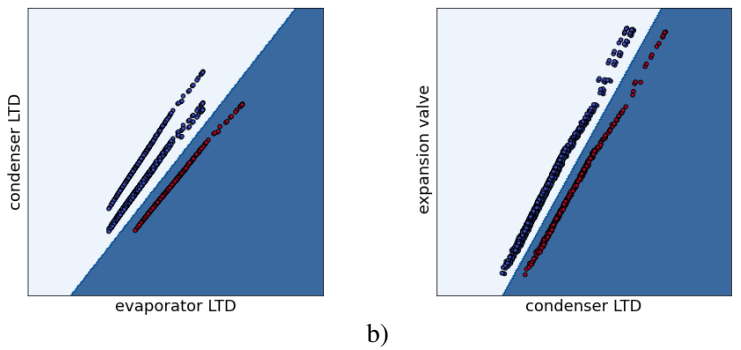

Figure 4.4 a) The separation of samples with (red) and without (blue) evaporator fouling using SVM with a linear kernel b) The separation of samples with (red) and without (blue) condenser fouling using SVM with a linear kernel.

Figure 4.4 shows plots for linear separation with the 2 most important features for fouling in evaporator and condenser in support vector machine. As discussed in 3.3, the straight line, seen in the plots, demonstrates that data for fouling in evaporator and condenser are linearly separable with these two chosen features and this classification task can be solved with a simple linear machine learning method. For charge loss, it was verified that three features were needed for linear separation to get an accuracy of more than 95%.

4.5 Simultaneous faults

The same 4 signals were then used to see if faults could still be isolated in the simultaneous fault dataset, displayed in table 2.2. The training data consisted of both the data from OAT dataset as well as the simultaneous faults one (samples 0-53, tables 2.1 and 2.2), while the test data consisted exclusively of the dataset with simultaneous faults (samples 30-53). Table 4.13 below, shows that SVM does not suffer any loss in performance and is able to detect and isolate these mixed faults. We can also see that although LR performs well on two of the faults, it is unable to meet the requirements for condenser fouling.

Table 4.13 Average performance scores for steady-state data for simultaneous faults, case 3: trained on 1 day, tested on 34 days, the 4 most important features.

Classifier	Model		
	SVM	**LR**	**Tree**
Evaporator			
Accuracy, %	100	100	83.5
False alarm rate. %	0	0	20
Condenser			
Accuracy, %	99.8	87.6	82
False alarm rate, %	0.3	13	19
Charge loss			
Accuracy, %	99.3	97	77.5
False alarm rate, %	0.02	1.9	22

4.6 Summary

Table 4.14 presents a summary of the dataset and machine learning model with the best performance scores. The data used for training the classifiers was simulated from field data, where the temperature from the cooling tower (as one of the inputs to the chiller model) varied about 0.2-4 Kelvin. Steady-state data has enhanced the performance scores for all three models and was chosen to be the final version for the training process. Comparing the results for 3 cases, 1 day has shown to be enough for training the model and reaching the baseline. Support vector machine with linear kernel has demonstrated to be the most robust model for this particular dataset, meeting outlined KPIs: accuracy scores above 95% of the three investigated ones, false alarm rate below 5%, the lowest fault severity (see table 2.1 samples 1, 11 and 21), and according to high accuracy for all three classifiers, the isolation can be estimated being higher than 95%. After feature importance analysis, 4 of 29 signals were selected as the most important and data has been proven to be linearly separable. Exact values for latency for the tests with the simultaneous faults are not shown in the table as 24 different combinations of fault levels (mentioned in section 2.4) were used.

Table 4.14 Summary for the final dataset and machine learning model

Chiller model	****		
Field conditions	8 hours from summer (July), input temperature variation 0.2-4 K		
Data source/ type	simulation from dynamic Chiller model		
Sampling time	15 min		
Amount of signals/ features	4 of 29 signals / mean values		
Training data	1 day, 1h window size for data sample steady state		
Faults	Fouling in evaporator Fouling in condenser Charge loss		
Machine learning model	Support vector machine (linear kernel)		
Test dataset	One-at-Time faults		
KPIs/ Classifier	Evaporator	Condenser	Charge loss
Accuracy, %	100	99.5	95.5
False alarm rate, %	0	0.6	0
Latency	1.56	1.27	0.993
Isolation	more than 95 %		
Test dataset	Simultaneous faults		
KPIs/ Classifier	Evaporator	Condenser	Charge loss
Accuracy, %	100	99.8	99.3
False alarm rate, %	0	0.3	0.02
Latency	combinations of fault severity levels		
Isolation	more than 95 %		

5

Discussion

5.1 Performance of the classifiers

According to the results with complete data, a significant difference is noticed in performance between these three machine learning models. SVM does generally perform the best and is least reliant on the size of the training data. Even with a single period of training data, SVM only fails the requirements for the charge loss fault as can be seen in table 4.4. Meanwhile, the decision tree model performs the worst and quickly decreases its performance when training data becomes limited. Another observation is made about a significant difference in the KPIs between the three faults. Generally, fouling in both evaporator and condenser is easier to detect than charge loss. This can either be because of that the signals observed are not sensitive to charge loss, or because the severity of the different faults is not similar in their relative magnitude. As shown in table 2.1, the severity levels differ in magnitude between the faults, which possibly causes variations in difficulties to detect them. Having said that, detection of the least severe case (samples 1, 11, 21 from table 2.1) was possible for every fault.

Improved KPIs can be seen with steady-state data for all models. This is likely because of the way the features are calculated as only mean values of small windows are studied separately. Windows of the same signal from steady state and the transient will be very different and make detection of low severity faults more difficult.The removal of the 3rd and 4th cumulants do not seem to have any significant negative effect on the KPIs. This is not surprising as the only fault induced change in any signal's steady state is a stationary offset, which would not be observable using those features, making them obsolete. In general, the same trend continues even here, SVM seems to perform the best and decision tree the worst. We can see that on average both LR and SVM fulfil the outlined requirements even with a single day of training data.

In section 4.4, it is shown that 4 signals is enough to detect and isolate the selected faults. Again, both LR and SVM fulfil the requirements (with 5 days of training data) while the decision tree performs significantly worse. For 1 day of training data only SVM could achieve the scores above minimum KPIs. After changing

regularization to L1-regularization, LR succeeded even for 1-day training data. Furthermore, figure 4.4 shows that as few as two signals are sufficient for detection of fouling in both the evaporator ("evaporator LTD" and "condenser LTD") and the condenser ("expansion valve" and "condenser LTD"), while charge loss requires at least 3 ("condenser LTD","subcooling" and "expansion valve"). The plots also show that this problem is linearly separable explaining the high KPI values when using simple linear models. We believe though, that this is only a symptom of the provided dataset (which is further discussed in 5.3) and the complex problem that Carrier is facing can not be solved that easily.

Furthermore, section 4.5 for simultaneous faults shows that SVM is able to isolate the faults. The simultaneous fault data shown in table 2.2 consists of one fault being either at nominal level or the lowest severity while varying the other two from low to high. It seems that no combination of two faults has the same effect on the selected 4 signals as the third fault, for example, charge loss together with condenser fouling does not have identical symptoms as evaporator fouling. This problem is still linearly separable as can be seen by the high accuracy results in table 4.13. SVM results demonstrate a marginal improvement with this test dataset, which is assumed to be a result of a larger training dataset as well as that the faults at much higher severity levels (than in the case with OAT tests) are also included now in the test data.

5.2 Dataset

Most of the time of the work in the thesis was spent on analyzing and trying to understand the data. The first weeks were dedicated to discussions about which chiller models would be used, what faults should be simulated and how much data was reasonable to have for training a machine learning model. While setting data requirements, the dataset has gone through a number of iterations. This was partly because of a model change, the introduction of additional signals, or bug fixes in the simulations. Verification of simulated data is quite challenging as it requires a good understanding of the chiller and despite some gathered knowledge about a chiller's basic functioning, we could not be sure that the data, we worked with, accurately represented data from a real chiller.

Another uncertainty is the implementation of faults in the simulations. Carrier does not have a lot of experience simulating faulty data and the models themselves are not made for that purpose. Although there are dynamic changes in the transient during chiller start-up, the faults mostly result in static offsets for different signals. This is also one of the reasons why the focus shifted towards the analysis of steady-state data as well as the removal of higher-order cumulants.

As mentioned in section 2.4, the incoming water temperatures were fetched from a real chiller to be used as inputs to the chiller model. This induced some realism into the simulated data and gave us some day-to-day variation. This variation,

however, was very minimal, as most of the days came from the same month, and all but 4 are during summertime. Additionally, all of the data was from one chiller and there were no variations in climate or capacity in the data. This means that the algorithm may be rather over-fitted to this specific data. The algorithm may succeed in the detection and isolation of faults for this specific chiller but would suffer big losses in performance if tested on at another time of year or another chiller. However, some thought has gone into trying to make the algorithm more generic and applicable. Such as, for example, the use of the LTD signals which are differences between two other signals, and therefore the steady-state data does not vary a lot with the load.

It has to be mentioned that there are scenarios where data from a healthy system may be classified as fouling. For example, the LTD signals that we use in classification of each fault will increase in value when the water flow to the heat exchanger is decreased. This will result in similar symptoms to fouling and would likely trigger an alarm. However, as these situations are not present in the dataset, they are not considered by the machine learning model. Consequently, the false alarm rates presented in the results are likely to increase significantly if the FDD algorithm is to be applied to a real chiller.

Due to time constraints, the algorithm's sensitivity to noise was not investigated, but it is not unlikely that the performance may decrease with the introduction of noise. We believe however that, because of how the algorithm is implemented, the effect that white measurement noise has on the features can be minimized. By increasing the window size and/or sampling frequency, the mean value of white noise should approach zero as the amount of data points per window increases. On the other hand, other types of noise, like bias in sensor due to some drift of calibration, may require more complex and non-linear machine learning models to solve this problem.

5.3 Machine learning models

The decision tree was chosen as the first pilot model as being one of the most interpretable machine learning methods. This method was trained first on data, simulated from another type of the chiller, and it was not determined yet if it would be the final version of data. After the performance scores did not meet the baseline on that particular training dataset, it was decided to test another supervised machine learning method - random forest. Random forest is an ensemble method, that uses multiple trees for prediction. The output is then selected by a majority vote of the trees' output during training. When decision trees become deeper, they tend to overfit, meaning high variance. In this case, random forest should reduce variance by building a bunch of trees using just some part of the training dataset for each tree. Most of the ensemble methods usually can significantly boost the performance of the model, but they become more like black boxes. Due to the loss of interpretability and also

rather high demand for computational time, the random forest was excluded as one of the possible machine learning methods.

As the problem formulation refers to a multiclassification task, four different classes were defined, representing a normal operation, fouling in the evaporator, fouling in the condenser, and charge loss. After training the model on some days from data, the first tests were done on another day, that the model has never seen before. A confusion matrix was used for performance evaluation and in the beginning, it had four different classes on both axes. Two KPIs, accuracy and false alarm rate, could be easily estimated with the help of Scikit learn functions and from the confusion matrix. The problem occurred with the isolation metric as having only one classifier for different faults did not allow us to see if the faults were isolated from each other. At that point, the idea of using three classifiers, each for certain fault detection, was developed. Multiclassfication problem transformed into three binary classification problems. While data simulation was still in the development stage, literature research was done on machine learning algorithms in order to find the most suitable ones for this task. According to several studies of fault detection, using the machine learning approach, one of the most common and powerful tools for pattern recognition was found to be support vector machine [Han et al., 2011]. It has been shown that SVM is a rather simple and robust model for fault detection problems. Moreover, SVM uses less memory because it stores only support vectors (that are formed by only part of the data) to make decisions and therefore it is faster. The performance of SVM model, constructed with a linear kernel, should give comparable results to logistic regression because the last one is also a linear classifier. SVM can also be extended with a non-linear kernel function if the problem occurs to be non-linear. The decision tree was noted to be prone to a decrease in performance for small training datasets. Still, it was decided to have this model as the third one in order to compare the performance scores with the other two parametric machine learning models: logistic regression and support vector machine.

5.4 Feature selection

The LMTD signals were believed at first to be important inputs to the FDD algorithm, but after some initial testing, it was clear that its response to the faults was very similar to that of the corresponding LTD signal. While LMTD signals require some relatively complex calculation as can be seen in section 2.3, the LTD is a simple subtraction and is already being calculated on-site for diagnostic/control purposes. This has led us to make the decision to only use LTD in the reduced signal list instead. However, this similarity in signal behavior may be a characteristic of this specific chiller system and may not be applicable to different chillers. In that case, LMTD can be used as a general method. Other signals that have very similar behavior to the LTD signals are refrigerant temperatures at different parts of the chiller. These are called "T suction" and "T condensing" and are noted as important

features in tables 4.2-4.8. When simulating data, the setpoint i.e. the leaving water temperature is held constant and the LTD becomes refrigerant temperature minus a constant value. But this is only a symptom of the way that the data was generated and should not be applied to a real system.

Changes in the opening of the expansion valve are also an indicator of variations in refrigerant flow and this might serve as a sign of fouling presence in the evaporator: when suction pressure drops, the expansion valve will open more to allow more refrigerant to the evaporator.

Another replacement of signals for the purpose of practicality was made in the subcooling case. As seen in tables 4.6-4.8, the most important features for charge loss include a signal named "cond.rellevel", which stands for the level of refrigerant in the condenser. This signal, however, is both expensive and difficult to measure accurately in a real chiller, as the refrigerant level in the condenser is often disturbed by the falling droplets of the condensing refrigerant. Instead, subcooling signal is used, as it is dependent on the number of wet tubes in the condenser (described in section 2.3), which gives us indirect information about the approximate level of refrigerant there.

6

Conclusion and suggestion on future work

6.1 Future work

As discussed in section 5.3, we consider the dataset to be the biggest uncertainty. All the samples in the dataset come from the same chiller meaning that FDD algorithm based on such data is unlikely to work on every other chiller, especially because they are usually built to order. Therefore, similar work should be done on a broader dataset, one that has more variance and can more accurately describe the problem that Carrier is facing. Such a dataset would have to include noisy data from chillers that differ in parameters, location, and the season they operate in to represent the variation of different units as well as their load. Generating a dataset would undoubtedly be time-consuming, but it probably could be one way to ensure FDD algorithm's applicability to real chillers.

We believe that there is still a lot of work to be done by analyzing the transient part of the data i.e. the chiller start-up. As faults seem to induce some change in the transient of the signal, it may be advantageous to train the algorithm on the dynamics of the transient instead of the of steady-state. If successful, this approach may provide FDD algorithm that is less sensitive to the operating conditions of the chiller. For a short time period we have attempted to implement this by calculating and using the covariance between every signal of the same window as features (as mentioned in section 3.1), but ultimately went with steady-state analysis after being unable to produce satisfactory results. However, this result is in no way definitive and further study with more sophisticated methods should be performed.

6.2 Conclusion

In the end, we were successful in implementing an FDD algorithm that fulfills the requirements outlined in section 1.2. The developed algorithm consists of three binary identifiers, one for each fault: fouling in the evaporator, fouling in the con-

denser, and charge loss (refrigerant leakage). It was shown in chapter 4 to achieve an accuracy score of above 95%, less than 5% false alarm rate as well as having low latency. The following conclusions can be drawn as the answers to the questions asked in section 1.2 :

- Dataset that was used in the thesis was shown to be linearly separable, and simple linear models could successfully be applied to solve the task.
- SVM performed the best and was the most consistent machine learning model out of the three. LR was comparable in performance, but a little more sensitive to small training datasets, while the decision tree was generally underperforming.
- The number of signals required for isolation of these faults was reduced to just 4.
- On average as little as a single day of operation (approximately 8 hours of steady-state data) was shown to be enough training data for SVM to reach the requirements.

These results, although promising, should not be accepted as the solution to the problem that Carrier is facing: developing a universally applicable FDD algorithm. The authors are skeptical that the dataset, which this thesis was based on, represents the full scale and the complexity of the problem. The situation out in the field is very complex, mainly because chillers are made and set up to order. This means that the different parameters can vary widely between different chillers, and that any FDD algorithm developed only considering a single chiller is unlikely to perform well. Therefore, it is impossible for us to decisively answer the questions concerning the broader problem. Nevertheless, we believe that some results from this smaller, less complex problem can still be of use. The 4 final signals were all chosen with the physics and ease of measurement in mind, and therefore should be applicable to chillers of similar type. Additionally, the performance evaluation of machine learning models should still apply, and we think that SVM should serve as a base for continued work in this area. SVM has shown to be a quite robust model and with the help of the kernel functions, it can also solve non-linear problems, that are likely to occur with future dataset iterations. To conclude, it is our hope that this thesis has provided some degree of useful insight to the team at Carrier and that it can serve as a good foundation for future studies.

Bibliography

Ardsomang, T., J. Hines, and B. Upadhyaya (2013). “Heat exchanger fouling and estimation of remaining useful life”.

Han, H., B. Gu, T. Wang, and Z. Li (2011). “Important sensors for chiller fault detection and diagnosis (fdd) from the perspective of feature selection and machine learning”. *International Journal of Refrigeration-revue Internationale Du Froid - INT J REFRIG* **34**, pp. 586–599. DOI: 10.1016/j.ijrefrig.2010.08.011.

Liang, L., E. Russell, and R. Braatz (2001). *Fault Detection and Diagnosis in Industrial Systems*. Springer.

Lindholm, A., N. Wahlström, F. Lindsten, and S. T.B. (2022). *Machine learning.A First Course for Engineers and Scientists*. Cambridge University Press.

Mohri, M., A. Rostamizadeh, and A. Talwalkar (2018). *Foundations of Machine Learning, second edition*. Massachusetts Institute of Technology.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). “Scikit-learn: machine learning in Python”. *Journal of Machine Learning Research* **12**, pp. 2825–2830.

Schölkopf, B. (2002). *Support Vector Machines and Kernel Algorithms*. RSISE, Australian National University.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
June 2022

Document Number
TFRT-6167

Author(s)
Vilius Koegst
Tatiana Orlova

Supervisor
Clas Jacobson, Carrier
Johan Grönqvist, Dept. of Automatic Control, Lund University, Sweden
Bo Bernhardsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
CHILLER DIAGNOSTICS Machine learning approach Carrier

Abstract

Chillers are large and complex machines that are used for temperature regulation in large buildings and plants. An undetected fault in the machine can lead to extended downtime and cause both great financial losses and increased environmental impact. Therefore Carrier, as manufacturer of chillers, is interested in developing fault detection algorithms capable of detecting the faults early so that they can be addressed before complications can occur. The faults, studied in this thesis, are refrigerant leakage, fouling (accumulation of unwanted material on a surface) in the evaporator and fouling in condenser, some of the main components in the machine.

The thesis aims to implement a fault detection and diagnostic algorithm (FDD), using supervised machine learning methods, and provide accurate and reliable identification of chosen faults. FDD is done by creating three binary classifiers, one for each fault, and by training the classifier on a labeled dataset, simulated from a highfidelity chiller model. Inputs for data generation in the chiller model come from the field data from 35 days in summer. Machine learning models - decision tree, logistic regression, and support vector machine, were used to develop the FDD algorithm and their performance was compared with 4 key indicators: accuracy, false alarm rate, latency, and isolation.

The results have shown that the problem occurred to be linearly separable, with respect to this particular dataset. Support vector machine model has achieved the highest scores in the detection of all three faults with the lowest fault severity level. Refrigerant leakage has proven to be the most difficult fault to detect, partly because the fault severity was almost close to normal operating conditions. Moreover, it was estimated that three signals were needed to detect leakage, while fouling could be separated by only the two most important signals. Finally, the transient data was removed and the training dataset was reduced to 1 day, which proved to be enough to detect and isolate the selected faults.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-61

Recipient's notes

Security classification

http://www.control.lth.se/publications/