Degree Project in Information and Network Engineering

Second cycle, 30 credits

Predictive Maintenance for Air-Conditioning Refrigeration and Heat Pump Systems

Data-driven Condition Monitoring and Fault Detection for Compressors

CI SONG

Predictive Maintenance for Air-Conditioning Refrigeration and Heat Pump Systems

Data-driven Condition Monitoring and Fault Detection for Compressors

CI SONG

Master's Programme, Information and Network Engineering, 120 credits
Date: June 11, 2024

Supervisors: Cheng Yang, Klas Berglöf
Examiner: Tobias Oechtering
School of Electrical Engineering and Computer Science
Host company: ClimaCheck AB
Swedish title: Prediktivt underhåll för luftkonditionerings-, kyl- och värmepumpsystem
Swedish subtitle: Datadriven tillståndsövervakning och feldetektering för kompressorer

Abstract

Heat pump and refrigeration systems consume a large amount of energy, and faults in these systems can cause significant wasted energy, so it is important to perform predictive maintenance for them. We propose a predictive maintenance approach that monitors isentropic efficiency in compressors based on historical data and machine learning. The results of monitoring can be used to detect faults in the systems.

We predict the isentropic efficiency and compare it with the measured value to detect a fault, which is viewed as the difference between prediction and measurement. The difference implies that the current pattern of the system operation is different from the pattern of the time when historical data were collected. Isentropic efficiency is selected as an indicator that characterizes the operating conditions of compressors, using suitable feature variables for isentropic efficiency prediction. Real data of representative systems are selected for validation. Different structures and optimizers of neural networks are designed and validated. Results show that such models can accurately predict the isentropic efficiency and exhibit better performance than linear models and support vector regression. The problem of an imbalanced dataset is also studied.

A Gaussian Mixture Model is used to analyse the distribution of historical data. A method is proposed that decides whether the model should be updated with new data when the distribution of input data has changed. It is assumed that if a proportion of new data proves that the system is fault-free and the rest of the data follow a different distribution and thus cannot provide persuasive results, the system is fault-free for the whole period. The model of one system is also applied on another similar system, in order to investigate the generalization performance of the model.

Keywords

Condition Monitoring, Fault Detection, Heat Pump and Refrigeration System, Neural Network, GMM

Sammanfattning

Värmepumpar och kylsystem förbrukar en stor mängd energi, och fel i dessa system kan orsaka betydande slöseri med energi, så det är viktigt att utföra förebyggande underhåll för dem. Vi föreslår en metod för prediktivt underhåll som övervakar isentropisk effektivitet i kompressorer baserat på historiska data och maskininlärning. Resultaten av övervakningen kan användas för att upptäcka fel i systemen.

Vi predikterar den isentropiska verkningsgraden och jämför den med det uppmätta värdet för att upptäcka ett fel, vilket ses som skillnaden mellan prediktion och mätning. Skillnaden innebär att det aktuella mönstret för systemdriften skiljer sig från mönstret vid den tidpunkt då historiska data samlades in. Isentropisk verkningsgrad väljs som en indikator som karakteriserar driftsförhållandena för kompressorer, med hjälp av lämpliga egenskapsvariabler för isentropisk effektivitetsförutsägelse. Verkliga data från representativa system väljs ut för validering. Olika strukturer och optimerare av neurala nätverk designas och valideras. Resultaten visar att sådana modeller kan förutsäga den isentropiska effektiviteten och uppvisa bättre prestanda än linjära modeller och stödja vektorregression. Problemet med ett obalanserat dataset studeras också.

En Gaussisk blandningsmodell används för att analysera fördelningen av historiska data. En metod föreslås som avgör om modellen ska uppdateras med nya data när fördelningen av indata har förändrats. Det antas att om en andel nya data visar att systemet är felfritt och resten av data följer en annan fördelning och därmed inte kan ge övertygande resultat, är systemet felfritt under hela perioden. Modellen för ett system tillämpas också på ett annat liknande system, för att undersöka modellens generaliseringsprestanda.

Nyckelord

Tillståndsövervakning, feldetektering, värmepump och kylsystem, neurala nätverk, GMM

Acknowledgments

I would like to thank Tobias Oechtering for managing the whole project and providing valuable suggestions.
I would like to thank Klas Berglöf for his expert knowledge about HVAC systems.

I would like to thank Saikat Chatterjee for his lectures in data science and machine learning.
I would like to thank Cheng Yang for his guidance, which I benefit a lot from.
I would like to thank my grandfather. He's an accountant, and does not talk much. He and grandmother raised three children and took care of the elderly with their low salary. He also helped raising grandchildren, including me. He usually cooked delicious and healthy food for me in my childhood.

Stockholm, June 2024
Ci Song

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

BIC	Bayesian Information Criterion
DBSCAN	Density-based Spatial Clustering of Applications with Noise
ELM	Extreme Learning Machine
EM	Expectation Maximum
GAN	Generative Adversarial Network
GD	Gradient Descent
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HVAC	Heating, ventilation, and air conditioning
iid	independent and identically distributed
LSTM	Long Short-term Memory
MAP	Maximum A Posteriori
MBGD	Mini-batch Gradient Descent
ML	Maximum Likelihood
MSE	Mean Square Error
PCA	Principal Component Analysis
pdf	probability density function
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
SDG	Sustainable Development Goal
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
UN	United Nations

List of Symbols Used

The following symbols will be later used within the body of the thesis.

Chapter 1

Introduction

1.1 Background

In buildings, air-conditioning, refrigeration and heat pump systems consume almost half of the total energy [1], and earlier fault detection can reduce energy waste of systems and is critical to Sustainable Development Goal (SDG) of United Nations (UN). However, heat pump and refrigeration systems have different characteristics due to brand, design, installation, and configuration, so it is hard to perform manual fault detection for large numbers of systems. The thesis project performs predictive maintenance for heat pump and refrigeration systems, and it monitors the change in the isentropic efficiency of compressors and detects faults based on monitoring results. It can utilize historical data to make automated fault detection and help engineers to judge the conditions and make decisions.

The project is conducted in cooperation with ClimaCheck Sweden AB, which provides performance analysis and monitoring services of Heating, ventilation, and air conditioning (HVAC) systems. It monitors HVAC systems worldwide and has access to some information and parameters of the systems.

Traditionally heat pump and refrigeration systems are relatively low-cost initially but cost much more during maintenance later. This leads to energy waste and downtime, and weakens the motivation of the industry to reduce failures. Nevertheless, the advancement of technology, increasing cost of energy, and awareness of sustainability are changing the market, and equipment owners focus more on sustainability reporting, energy efficiency, and reduced downtime, which boosts the demand for predictive maintenance and automated fault detection with historical data.

1.2 Problem

The problem the thesis studies is to perform automated fault detection for compressors in heat pump and refrigeration systems, and it involves the following questions.

1. How to use historical data of a system to detect faults in the same system? What if there are insufficient data or data changes over time?

2. Can historical data of a system be used to detect faults in another system? If not, can we at least provide some information as a reference?

To solve the problems above, we need to investigate the generalization of the model. In machine learning, generalization refers to the feature that a model trained by a dataset fits into new data. The first is to generalize to another input data distribution. The second is to generalize to another heat pump and refrigeration system, which can involve both generalization of data distribution and new mapping from input to output.

1.3 Goals

The goal of this project is to perform predictive maintenance for heat pump and refrigeration systems. This has been divided into the following sub-goals:

1. To identify the variable that can better depict the operating condition of compressors in heat pump and refrigeration systems.

2. To select the features and model that can predict the fore-mentioned variable.

3. To understand how input data of the model change with time and try to update the model accordingly.

4. To test how one system's model can be applied to similar systems.

5. To perform fault detection based on results from previous goals.

1.4 Challenges

There exist some challenges in the design of automated fault detection schemes:

1. There can be insufficient data, especially for new devices or devices that have changed the operating mode.
2. The heat pump and refrigeration systems are building and equipment dependent, and work under time-varying conditions [2].
3. We need to differentiate between faults and sub-optimal operating conditions.
4. We have to reduce false alarm rates.
5. We need to perform hyperparameter tuning for our models, and we can only perform this by experience.

1.5 Delimitations

We narrow the research problem down to compressors in heat pump and refrigeration systems because it is easier to locate the faults when detected, and this can also help us in more precise feature selection and modelling.

We only perform fault detection. Diagnosis is not included in the thesis. Fault in the thesis is defined as the difference between predicted and measured isentropic efficiency.

The feature selection for the model is conducted with expert knowledge from ClimaCheck Sweden AB, and tested with real data afterwards.

We use data from different real systems collected by ClimaCheck Sweden AB. Some of them are directly obtained from sensors, while others are calculated by the algorithm of ClimaCheck Sweden AB. Both a faulty system and a faulty sensor can cause a deviation in the model.

1.6 Structure of the Thesis

Chapter 2 presents relevant background information about the thermodynamics of compressors and methods in data science. Chapter 3 shows the methodology and method used to solve the problem studied in this thesis. Chapter 4 exhibits the data we use and numerical results. Chapter 5 concludes the thesis and discusses some future work.

Chapter 2

Background

2.1 Automated Fault Detection

Generally, most of the models listed in Table 2.1 are data-driven models and only a small proportion of them design physics-based models based on thermodynamics. Physics-based models are not suitable when real HVAC systems and their operating conditions are complex. Only some create physics-based models and analyse their parameters with data science technologies.

There are four common schemes for data-driven fault detection from papers discussed in Sections 2.1.1 and 2.1.2:

1. Regression: Predict some variable, and compare it with the observed one.

2. Classification: Label data as 'normal' or 'faulty' and classify training or test data with the model directly.

3. Encoder-decoder: Use an encoder-decoder architecture where only normal data can be reconstructed successfully. [3]

4. Transform: Transform-based methods first extract some features via transforms and then make the decision. For example, PCA-based methods choose the principal components of the data and judge whether they exceed the limit [4].

Regression and classification tasks can be realized by many kinds of models, including Support Vector Machine (SVM), Long Short-term Memory (LSTM), and other neural networks. These models can also be modified to enhance the performance, and the inputs can be measured directly or preprocessed through algorithms like Principal Component Analysis (PCA). The encoder-decoder method utilizes models like LSTM autoencoder and Generative Adversarial Network (GAN). The last method is achieved through PCA or other transforms.

Moreover, most methods perform time series analysis, where data from a time period become the input of the models. Context information can also be taken into consideration [5].

Some methods rely on expert knowledge for feature selection, while others use algorithms like PCA to select the most influential variables.

Most of the methods use synthetic data or data from a limited number of real buildings.

Some methods also achieve fault diagnosis besides detection, while others view diagnosis as part of future work.

2.1.1 Physics-based Methods

In physics-based methods, physics models are built and the parameters of the models are calculated. Usually, the physics models are also combined with data-driven models. If the output of the model exceeds a threshold, a fault is declared. Sun, *et al.* [2] apply two physical models for chiller and cooling tower respectively, and calculate the parameters of the models. After that, they predict the parameters in the future with Kalman filters and detect faults with Statistical Process Control (SPC), which means that a fault is detected when the parameter deviates too much from its normal range. The method is also modified during the detection of gradual faults, for the trade-off between low false alarm rate and high detection rate. Tests also show that it can detect both device faults and sensor faults. For faults that spread through different subsystems, they introduce coupling variables and process them as before. Bashi, *et al.* [17] introduced a multi-unit hybrid system model based on Gaussian-mixture (GM) distribution to mimic large-scale HVAC practical system. Parameters are obtained from expectation-maximization (EM) estimation, and faults are detected during hypothesis testing. The method is tested under different conditions. Takeuchi *et al.* [21] designed a scaling law-based model for refrigerant leakage detection independent of system configuration, which can avoid the problem of insufficient training data. The degree of leakage is derived from the physics model and a fault is detected when it exceeds the threshold. Yan *et al.* [25] propose a physics model and Hidden Markov Model (HMM) for cooling coil, and values of the physics model are used in the fault indicator matrix of HMM.

Table 2.1: Comparison of Related Works

Paper	Supervised	Type of Model	Model	Input Features	Source of Data	Diagnosis
[6]	Yes	Data-driven	isolation forest	Energy consumption	Simulation	1 Fault
[6]	Yes	Data-driven	One Class SVM	Energy consumption	Simulation	1 Fault
[6]	No	Data-driven	LSTM Autoencoder	Energy consumption	Simulation	1 Fault
[7]	Yes	Data-driven	SVM	Temperatures, Pressure	Simulation	5 Faults
[8]	No	Data-driven	Kernel PCA	Temperatures, Pressure, Electrical Power, Ventilation and Compression	Simulation and ASHRAE Dataset	1 Fault
[9]	Yes	Data-driven	neural network	Temperatures, Cooling Capacity	Simulation	5 Faults
[2]	Yes	Physics-based	Kalman Filter	Physics Model Parameters	Simulation of 1 building	1 Fault
[10]	No	Data-driven	LSTM Autoencoder	Temperature	Supermarkets	1
[11]	No	Data-driven	GAN	Sound	A refrigerator unit	1 Fault
[12]	No	Data-driven	SDAR	Temperature, Energy	Kaggle[13]	1 Fault
[14]	Yes	Data-driven	Federated Learning CNN	Temperature, Pressure, Water/Air Flow	ASHRAE Project [15] [16]	7 Faults
[17]	Yes	Physics-based	multi-unit hybrid system model	Temperature, Pressure, Air Flow	Simulation and real data	1 Fault
[18]	Yes	Data-driven	CNN	Temperature, Pressure, Flow Rate	Condensing unit	1 Fault
[19]	Yes	Data-driven	PCA-SVM	Temperature, Pressure, Flow Rate	Condensing unit	1 Fault
[20]	Yes	Data-driven	proximal supersphere-based SVM	Temperature, Pressure, Humidity	Air-conditioning unit of KLD-29	5 Faults
[21]	Yes	Physics-based	Scaling law-based model	Temperature	Real-world data	1 Fault
[22]	Yes	Data-driven	Bayesian Diagnosis Network	Temperatures	chiller in ASHRAE RP-1043	6 Faults
[23]	Yes	Data-driven	wavelet transform	Power	REFIT Dataset	1 Fault
[24]	No	Data-driven	LSTM	Temperatures, Pressure, Humidity	Real data	1 Fault
[1]	Yes	Data-driven	2-D CNN	Temperatures	Simulation data	1 Fault
[25]	Yes	Physics-based	HMM	Temperatures	Simulation and real data	4 Faults
[26]	Yes	Data-driven	Support vector regression	Temperature, Illumination, Humidity	Data from a building	1 Fault
[26]	Yes	Data-driven	DNN regression	Temperature, Illumination, Humidity	Data from a building	1 Fault
[26]	No	Data-driven	Autoencoder	Temperature, Illumination, Humidity	Data from a building	1 Fault
[27]	Yes	Data-driven	Support vector regression	Pressures, Temperatures, Frequency	VRF system	1 Fault
[28]	Yes	Data-driiven	Multi class SVM	Temperatures, Power	A building in London	1 Fault
[29]	Yes	Data-driven	ELM	Energy, Temperature, Humidity	ASHRAE 1312-RP	19 faults

2.1.2 Data-driven Methods

One way to perform fault detection is through classification, which is a supervised method. Van de Sand, *et al.* [7] listed 26 different features from vapor compression refrigeration systems, including different temperatures, pressures, efficiencies, and heat flows, and chose the five most important ones as the input of model with adaptive genetic algorithm (AGA). The SVM with one-vs-rest approach for diagnosis of different faults. The ROC curve exhibits that the method can diagnose oil-retentions better. Guo, *et al.* [9] trained a BP neural network model with particle swarm optimization for fault prediction in cooling systems, which is proved to be better than merely the BP neural network. Ke Yan *et al.* [29] use Kalman filter to generate state vector from measurements, and use Extreme Learning Machine (ELM) to make fault diagnosis. Only normal data are used for training Kalman filter and both normal and faulty data are used for training ELM. The proposed ELM is modified and achieves better performance than other methods. Wang, *et al.* [14] designed Federated Learning-based CNN for fault detection and diagnosis in chillers and air handling units (AHU) jointly, with better performance in less serious faults. The feature selection is based on previous studies. Soltani *et al.* [18] trained a CNN with data from the condensing unit of supermarket refrigeration system, where superheat temperature, suction pressure, compressor speed, and compressor mass flow rate are selected as features. Tests are done on different sampling rates, data log lengths, and noisy data. In another research of Soltani *et al.* [19], PCA is first used to select influential features, and the features are then sent to SVM for classification. The PCA-SVM algorithm has good robustness against noise and different running modes. Elnour *et al.* [1] proposed 1-D-to-2-D data transformation approach to train data with 2-D CNN, which exhibits better performance than other methods. Different CNNs are also tested. Dey *et al.* [28] proposed a feature extraction method before modelling, and then SVM is used to classify the data of features. Liu *et al.* [20] trained proximal supersphere-based SVM to detect five different types of faults, during which except sorting out parts (ESOP) scheme enhanced the performance. The experiment is conducted on train air-conditioning units. Zhao *et al.* [22] designed a Bayesian Diagnosis Network for fault diagnosis, where the fault symptom layer receives inputs and the fault layer generates outputs. The framework can integrate useful information from different sources and can be tolerant of uncertainties due to probabilistic analysis and graph theory. Moreover, Chew *et al.* [30] collected more prior knowledge for additional information layer.

Besides classification, regression methods can also be used to perform fault detection. If the calculated value deviates too much from the measured one, a fault is declared. Yokouchi *et al.* [24] collect data from railway vehicles and predict air blowing temperature with LSTM model. Then they measure the difference between predicted and actual blowing temperature for fault detection. Guo *et al.* [27] proposed a support vector regression method to predict subcooling parameter and detect faults in variable refrigerant flow (VRF) systems, exhibiting better performance than the multiple linear regression model.

Encoder-decoder based methods use structures that can reconstruct normal inputs. This is an unsupervised method and only uses normal data to train the model. Faults are declared when data cannot be reconstructed. Alghanmi, *et al.* [6] made energy-based fault detection with three approaches - isolation forest, One Class SVM and LSTM Autoencoders. Data collected under different fault severity are used and LSTM Autoencoders exhibit best performance. Mayer, *et al.* [10] collected data from refrigeration systems of supermarkets and trained LSTM Autoencoders, which predicts future inputs according to previous ones. If the sensor data are vastly different from the prediction, the time series will be viewed as faulty. The work proves the validity of such ML models and raises further questions, like choice of threshold and integration with other methods and expert knowledge. Hatanaka, *et al.* [11] utilized efficient GAN [31] to perform anomaly detection on sounds collected by microphone, and its performance exceeded autoencoder and variational autoencoder. The generator of GAN attempts to generate an image similar to the input, and this can not be achieved for abnormal data, because only normal ones are used for training. The heat map exhibits the difference between the input image and the generated image, which can be utilized for anomaly detection. Wang *et al.* [26] tested three methods for the prediction of electrical load or fault detection: Support vector regression, deep neural network (DNN) regression, and autoencoder with and without reconstruction. Among them, support vector regression has the worst performance and works as a benchmark. For autoencoders, anomaly score is also introduced.

Transform-based methods select some features of the result. If it exceeds the threshold, then a fault is declared. This is similar to the regression method because they all compare the result with the threshold, but PCA based methods perform PCA to input features first. Simmini, *et al.* [8] developed a self-tuning Kernel PCA method to predict if the system is normal or faulty, exhibiting advantage over linear PCA, One Class SVM and isolation forest. The fault is detected when SPE score [32] exceeds the threshold. However, further works are required when the operative conditions of the system change. Gupta, *et al.* [12] assumes normal systems are stationary, so they use sequentially discounting auto regression (SDAR) to find changes in time series from Kaggle [13] and calculate an outlier score. Lemes *et al.* [23] perform wavelet transform on active power series from devices, and the result is used for fault detection. The method can generate the beginning and end instant of faults and presents low computational complexity.

2.2 Thermodynamics of Compressors

Compressors can be classified into rotary-vane, scroll, and reciprocating-piston [33], and they have some different characteristics. In air-conditioning refrigeration and heat pump systems, they all operate under heat pump and refrigeration cycles, and their isentropic efficiency can be calculated based on the cycle.

2.2.1 Heat Pump and Refrigeration Cycle

Vapour compression cycle is an implementation of the reversed Carnot cycle, where energy is transferred from cold to hot phases. Thus, this can be used for heat pump and refrigeration. Figure 2.1 illustrates a simplified model of the difference between practical and ideal cycles, and we only take into account the influence of the compressor, which relates to the main research problem of the proposed method. The x and y axes represent enthalpy and pressure. The thin curve represents the properties of the refrigerant, and its left and right halves are saturated liquid and saturated gas respectively. In the cycle depicted by the thick curve, point 1 represents suction gas, which is the inlet of the compressor, and its pressure is P_L. In contrast, point 2 represents discharge gas, which is the outlet of the compressor and superheat, and its pressure is P_H. Non-isentropic vapour compression is from point 1 to 2, whereas isentropic vapour compression is from point 1 to 2', which means that the entropy remains unchanged. Isobaric sub-cooling and condensation is from 2 (2') to 3, isenthalpic liquid expansion is from 3 to 4, and isothermal and isobaric evaporation is from 4 to 1.

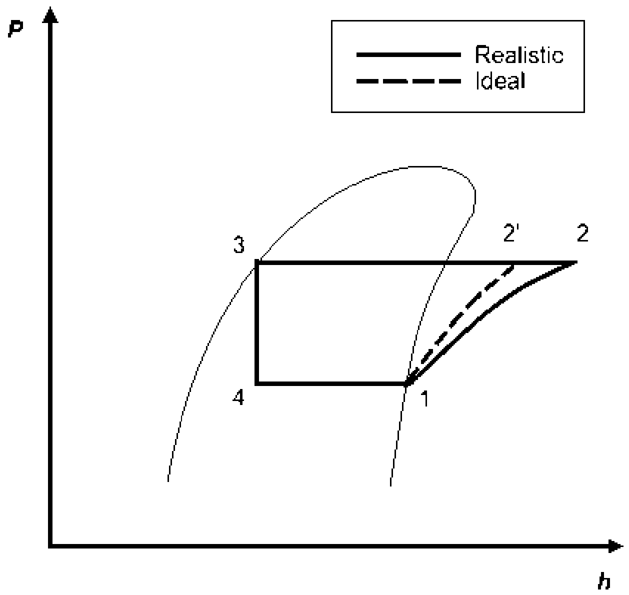


Figure 2.1: Vapour Compression Cycle [34], where Realistic Compression is Added

2.2.2 Enthalpy and Isentropic Efficiency

The enthalpy H of a thermodynamic system is defined as the sum of its internal energy and the product of its pressure and volume [35], and we use $h = H/m$ for further calculation, where m is the mass of the system. Usually, we know h from the property table of the used refrigerant, given temperature and pressure. If the refrigerant is saturated, only one of them is required. REFPROP [36] is a software that has stored such values in tables.

Isentropic efficiency measures the performance of the compressor. It is the ratio between work required in an isentropic process and actual work [37] and is given by

$$\eta = \frac{h_{2'} - h_1}{h_2 - h_1}, \tag{2.1}$$

where the subscripts of enthalpy h correspond to the point notations in Figure 2.1. In the rest of the thesis, the unit of isentropic efficiency will remain percent (%).

In practice, we can use the ten-coefficient polynomials [33] to calculate isentropic efficiency. The compressor polynomial regression coefficients are usually provided by manufacturers, although this may not always be accurate.

The work [38] exhibits the relation between isentropic efficiency and pressure ratio between P_H and P_L for different systems. The curves vary for different versions of devices. This also proves the demand for a data-driven method of isentropic efficiency prediction.

2.3 Clustering

Clustering methods classify data into different clusters. This can help us to understand how data changes over time.

2.3.1 Gaussian Mixture Model

To approximate complex probability densities, we can use a linear combination of a basic distribution, which is known as mixture distributions. Gaussian Mixture Model (GMM) [39] is a linear combination of Gaussian distributions

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.2}$$

where each Gaussian density $N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is called a component, $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ are mean and covariance matrix of the corresponding component, and π_k is named mixing coefficient.

From the sum and product rules, the marginal probability density is

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(k)p(\boldsymbol{x}|k). \tag{2.3}$$

If we view $\pi_k = p(k)$ as the prior probability of picking the k-th component, and $p(\boldsymbol{x}|k)$ as the component, then Eq. 2.3 and Eq. 2.2 exhibit the same relation. Thus, we can express GMM in terms of a discrete latent variable $\boldsymbol{z} = [z_1, \ldots, z_K]$ where $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$,

$$\begin{aligned} p(\boldsymbol{x}) &= \sum_{\boldsymbol{z}} p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z}), \\ p(\boldsymbol{z}) &= \prod_{k=1}^{K} \pi_k^{z_k}, \\ p(\boldsymbol{x}|\boldsymbol{z}) &= \prod_{k=1}^{K} N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}. \end{aligned} \tag{2.4}$$

The number of components K can be determined from Bayesian Information Criterion (BIC) [40]

$$BIC(K) = K \ln n - 2 \ln p(\boldsymbol{x}|\boldsymbol{\theta}), \tag{2.5}$$

where $\boldsymbol{x}$ is a set of data points, n is the number of data points and $\boldsymbol{\theta}$ is optimal model parameters given K.

The optimal K is obtained when BIC is minimized

$$K_{opt} = \min_{K} BIC(K). \tag{2.6}$$

The parameters $\theta = \pi_1, \ldots, \pi_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$ can be calculated by the Expectation Maximum (EM) algorithm, which is shown in Appendix A. The EM algorithm returns the local optimal solution so the results can differ at different times.

2.3.2 DBSCAN

Density-based Spatial Clustering of Applications with Noise (DBSCAN) [41] is a clustering algorithm that groups nearby points and labels outliers. Algorithm 1 shows the processes of DBSCAN.

Algorithm 1 DBSCAN

Data: points, radius of neighborhood ϵ, minimum number of points in a cluster $minP$

Result: points are classified into clusters or noisy data

A point p is a core point if there are at least $minP$ points located within the distance of ϵ.

A point q is directly reachable from p if the distance between p and q is less than ϵ. It implies that p is a core point.

A point q is reachable from p if there exists a path from p_1 to p_N, where $p_1 = p$, $p_N = q$, and p_{n+1} is directly reachable from $p_n, n = 1, \ldots, N$

All the points not reachable from any other points are noisy points, and a core point forms a cluster with all the reachable points.

The function can be realized with the Python library sklearn [42], which also provides GMM. Compared with GMM, DBSCAN can mark noisy data when no suitable cluster is found, but it only returns which cluster a point belongs to without corresponding probability.

2.4 Regression Models

2.4.1 Linear Regression

Linear regression models view output as a linear combination of inputs. Assume $\boldsymbol{X} \in \boldsymbol{R}^{d \times n}$ is a dataset with n d-dimension data records, and $\boldsymbol{Y} \in \boldsymbol{R}^{q \times n}$ is the corresponding labels. The model is

$$f(\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{X}, \tag{2.7}$$

where $\boldsymbol{A}$ is parameters that can be calculated by least squares method [39] as follows

$$\begin{aligned} \boldsymbol{A} &= \operatorname{argmin}_{\boldsymbol{A}} \quad ||\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}||_F^2 + \lambda||\boldsymbol{A}||_F^2 \\ &= \boldsymbol{Y}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I})^{-1} \end{aligned} \tag{2.8}$$

with Frobenius norm $||\boldsymbol{A}||_F^2 = trace(\boldsymbol{A}^T\boldsymbol{A})$, $\boldsymbol{I}$ is identity matrix, and λ is the regularization parameter.

2.4.2 Support Vector Regression

Support Vector Regression utilizes SVM [43] to perform regression tasks. We first want to construct a linear SVM regression model

$$\begin{aligned} & f(\boldsymbol{x}) = \boldsymbol{\beta}^T\boldsymbol{x} + \boldsymbol{b} \\ \text{subject to} \quad & |y_n - f(\boldsymbol{x}_n)| \leq \epsilon, \quad n = 1, \ldots, N, \end{aligned} \tag{2.9}$$

where $\boldsymbol{x}$ is a vector of observations, y is a scalar of the target, and $\boldsymbol{\beta}$ and $\boldsymbol{b}$ are a weight vector and a bias vector.

However, not all the data can satisfy the constraints, so we introduce slack variables ξ_n and ξ_n^*. The optimization problem is then

$$\begin{aligned} \text{minimize}_{\boldsymbol{\beta},\boldsymbol{b}} \quad & \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + C\sum_{n=1}^{N}(\xi_n + \xi_n^*) \\ \text{subject to} \quad & y_n - (\boldsymbol{\beta}^T\boldsymbol{x}_n + \boldsymbol{b}) \leq \epsilon + \xi_n \\ & (\boldsymbol{\beta}^T\boldsymbol{x}_n + \boldsymbol{b}) - y_n \leq \epsilon + \xi_n^* \\ & \xi_n \geq 0 \\ & \xi_n^* \geq 0 \quad n = 1, \ldots, N. \end{aligned} \tag{2.10}$$

We can calculate the Lagrangian and dual function and then solve it using the Karush–Kuhn–Tucker (KKT) conditions if the constraint qualifications are satisfied.

We can also use the kernel trick [39] to modify the model $f(\boldsymbol{x})$ so that the input becomes $\phi(\boldsymbol{x})$ instead of $\boldsymbol{x}$. Through using the kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \phi^T(\boldsymbol{x})\phi(\boldsymbol{x}') \tag{2.11}$$

we do not need to obtain $\phi(\boldsymbol{x})$ explicitly. A common kernel is Radial Basis Function (RBF)

$$k(\boldsymbol{x}, \boldsymbol{x}') = e^{-\gamma||\boldsymbol{x}-\boldsymbol{x}'||^2}, \tag{2.12}$$

where γ is a free parameter that should be set by experience or tuned.

We can also use the Python library sklearn [42] to solve the support vector regression problems with RBF kernels.

2.4.3 Neural Network

A neural network is composed of multiple hidden layers and an output layer. The structure we use is inspired by fully connected layers in LeNet [44]. Each hidden layer takes the form

$$\boldsymbol{o} = ReLU(\boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b}) \tag{2.13}$$

and output layer follows

$$\boldsymbol{o} = \boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b}, \tag{2.14}$$

where $\boldsymbol{x}$ and $\boldsymbol{o}$ are input vectors and output vectors of the layer, and $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight matrix and the bias. The output of a hidden layer is the input of the next layer.

The Rectified Linear Unit (ReLU) is a type of activation function:

$$ReLU(x) = \begin{cases} x & x > 0, \\ 0 & \text{otherwise}, \end{cases} \tag{2.15}$$

Another activation function is the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{2.16}$$

The non-linearity of the activation function makes neural networks suitable for learning complex non-linear representations.

AlexNet [45] applies ReLU function and a technology called dropout [46], which randomly disconnects the connections while training to prevent overfitting. Overfitting is a phenomenon where the model has significantly better performance when using the training data than using the test data.

Batch normalization [47] is another technology that can enhance the performance of neural networks, and it modifies the means and variances of inputs in each layer.

2.4.4 Gradient Descent

Input data $\boldsymbol{x}_t$ passes through the whole network until the output is generated. This is called forward propagation.

In contrast to forward propagation, backpropagation involves calculating the gradient of the model through the chain rule, in which gradients are calculated from the back of the model to the front. This function is critical to the update of model parameters.

The loss function is a criterion that measures the error between outputs and labels. The proposed method uses Mean Square Error (MSE) as the loss function

$$l(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{y}_n - f(\boldsymbol{x}_n, \boldsymbol{\theta}))^2, \tag{2.17}$$

where $\boldsymbol{y}$ is the vector of target values, $\boldsymbol{\theta}$ is the vector with parameters of the model, and for a fully connected neural network $f(\cdot, \boldsymbol{\theta})$, $\boldsymbol{\theta}$ is $\boldsymbol{W}$s and $\boldsymbol{b}$s in Equations 2.13 and 2.14.

The loss function is supposed to be minimized for optimal parameters. There are different methods to achieve this [39]:

Gradient Descent (GD) method calculates the loss of the whole dataset and updates parameters by deducting the gradient of the loss function with respect to parameters from the old parameters. In this way, the parameters move towards the minimum loss function.

Stochastic Gradient Descent (SGD) performs GD to one random sample of the dataset at a time.

In Mini-batch Gradient Descent (MBGD), a dataset is divided into different batches. For each batch, the gradient of the loss is calculated and then utilized to update the parameters.

To reduce the effect of randomness, momentum is introduced when updating, as is shown in Algorithm 2. The applied gradient is a combination of the latest gradient and historical gradient. The momentum is usually set at 0.9. The process can be repeated many times until the loss function converges or the maximum epoch number is reached. Note that in PyTorch [48], SGD optimizer performs gradient descent, so it can also be used in MBGD with momentum.

PyTorch also provides Adam optimizer [49] as an alternative to gradient descent. The Adam algorithm is shown in Algorithm 3:

Algorithm 2 MBGD with momentum

Data: model parameters θ_0, learning rate lr, momentum α, loss function $l(\theta)$
Result: new model parameters θ_t

t=0
for *epoch number* **do**
  for *batches in dataset* **do**
    $t \leftarrow t + 1$
    $input, label \leftarrow batch$
    $output(\theta_{t-1}) \leftarrow model(\theta_{t-1}, input)$
    $l(\theta_{t-1}) \leftarrow MSE(output(\theta_{t-1}), label)$
    $g_t \leftarrow \alpha \cdot g_{t-1} + (1 - \alpha) \cdot \nabla_\theta l(\theta_{t-1})$
    $\theta_t \leftarrow \theta_{t-1} - lr \cdot g_t$

Algorithm 3 Adam algorithm

Data: model parameters θ_0, learning rate γ, weight decay λ, algorithms parameters β_1, β_2, loss function $l(\theta)$
Result: new model parameters θ_t

Initialization:
first moment $m_0 = 0$
second moment $v_0 = 0$
for *t=1 to T* **do**
  $g_t = \nabla_\theta f_t(\theta_{t-1})$
  $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
  $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
  $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$
  $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$
  $\theta_t = \theta_{t-1} - \gamma \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$

Chapter 3

Methods

3.1 Problem Description

The objective of the project is automated fault detection of air-conditioning, refrigeration and heat pump systems. Isentropic efficiency plays a key role in reflecting the condition of compressors in the systems, so we first need to perform condition monitoring for isentropic efficiency. However, the efficiency itself does not imply whether the system is faulty or normal. We compare the measurements and predictions of isentropic efficiency, and deviations between them are considered as faults.

For this purpose, we need to collect historical data when the system is fault-free, which has been implemented by the software of ClimaCheck Sweden AB. ClimaCheck Sweden AB can calculate isentropic efficiency with their measurements and algorithm, so the core target of the project is to predict what the efficiency should have been if the system behaves in the same way as what is recorded in historical data. After this, we only need to compare the difference between predicted values and measured values from ClimaCheck Sweden AB with a threshold, and automated fault detection is achieved.

If we want to perform continuous maintenance for the systems, then we also need to propose a method that updates the prediction procedure because the modes of the HVAC systems change over time. For example, a heat pump and refrigeration system can be air-conditioning in summer and heat pump in winter. This will be elaborated on in Section 4.4.

We also test whether one model trained for one system can be applied to another system. We will exhibit some initial results and empirical conclusions.

The proposed method in this thesis makes use of real data for both training and testing of our proposed models.

3.2 Problem Formulation

We make the following assumptions before formulating the problem:

1. Faults in the systems are exhibited in deviations between measurements and predictions of isentropic efficiency. When the model is representative enough but still cannot make accurate predictions of current isentropic efficiency, it is assumed that the system is faulty.

2. If we know that the system is fault-free for a long enough proportion of the time period, we assume that the system is fault-free for the whole time period and all the data can be used for an update of the model.

3. When the model of a system is applied to a similar system, a deviation between predictions and measurements is anticipated.

Assume that ClimaCheck Sweden AB has collected a historical dataset $A = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, y_1, \ldots, y_N\}$, where the vector of input features $\boldsymbol{x}_n = [x_{n1}, \ldots, x_{nd}, \ldots, x_{nD}]^T$, $x_{nd} \in \boldsymbol{R}$ and $y_n \in \boldsymbol{R}$ is the corresponding isentropic efficiency of $\boldsymbol{x}_n$. Data within a dataset are from different time points n. They are not independent and identically distributed (iid), but we treat them as iid. We design a model $f(\cdot)$ and train the model using A. For test data $\boldsymbol{x}_t$ and y_t, if $|y_t - f(\boldsymbol{x}_t)|$ exceeds the threshold ϵ, an alarm is raised. Usually, $\boldsymbol{x}_t$ and y_t should be from the same system as A is from. If not, the model $f(\cdot)$ may not be precise.

Parameters of model $f(\cdot)$ are obtained based on methods in Section 2.4. For example, we use the least squares method for linear regression, find support vectors for SVM, and minimize the loss function of neural networks. For hyperparameters, we choose optimal ones according to results in the validation set or use default values in algorithms of the Python library.

Assume that dataset B is obtained later than A. Given a new data record $\boldsymbol{x}$ from dataset B, we propose a $score(\boldsymbol{x})$ based on $\boldsymbol{x}$ and probability density function inferred from A to depict its relationship with A, which also reflects whether the prediction of isentropic efficiency should be accurate or not. If for most data $\boldsymbol{x} \in B$, $score(\boldsymbol{x}) \geq threshold$, and $|y - f(\boldsymbol{x})| \leq \epsilon$, we decide that data in B are all fault-free and combine A and B to retrain $f(\cdot)$.

Figure 3.1 exhibits the flowchart of the whole proposed method, in which ellipsoid, rectangle, and rhombus represent data, algorithm, and decision respectively.

The inputs of the models are data collected from the compressor systems. For example, PH and PL represent high pressure P_H and low pressure P_L,

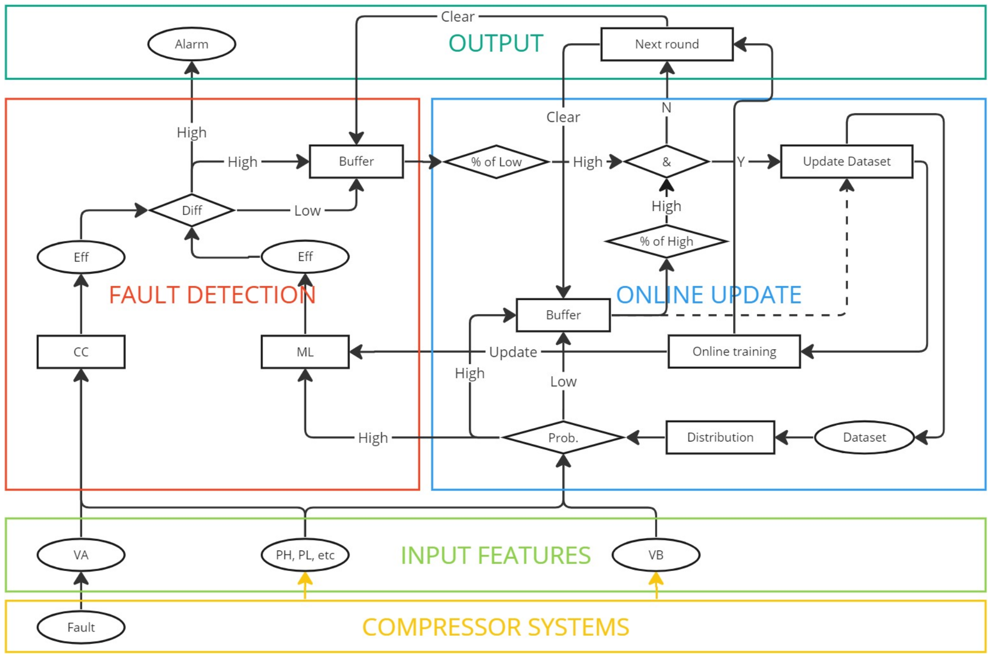


Figure 3.1: Flowchart of the Proposed Method

and they are the inputs of the algorithm of ClimaCheck Sweden AB and the proposed model. VA and VB are input variables chosen by the algorithm of ClimaCheck Sweden AB and the proposed model respectively.

The left part of the flowchart is the fault detection algorithm. CC represents the software of ClimaCheck Sweden AB, which generates measured isentropic efficiency. The proposed machine learning model is denoted by ML. ML is trained in initialization and updated during online training. If the difference between outputs Eff of ML and CC exceeds the threshold, then a fault is detected and an alarm is sent. The fault detection algorithm decides whether to raise the alarm for every time point so that it can process continuous input and provide engineers with when the fault occurs exactly.

The right module in the flowchart corresponds to online update. The training does not start until enough data are collected, so a buffer is required to save the data. To be specific, let us assume that we have a trained model and some data records from a new month. We first use the data records that highly fits the probability density function inferred from the old dataset to test, and the results are saved in the buffer, which is the function of the fault detection module. If a large proportion of the new data has a high probability and most of the test errors are low, we can assume that the system is not faulty in the new month, and all the data can be used for updating the dataset. Then the model is updated and buffers are cleared for the new round. The fault detection module can work without this update module.

Details and justification of the proposed flowchart will be elaborated on in the next few sections.

3.3 Feature Selection

The high pressure P_H and low pressure P_L are related to enthalpy h, which is used for the calculation of isentropic efficiency in Eq. 2.1, and our proposed method also utilizes them as inputs of the model. ClimaCheck Sweden AB selects variables closely related to the definition of isentropic efficiency to predict actual efficiency, while our proposed method chooses variables unaffected by potential faults in the systems. Our chosen variables can contribute to the prediction of the isentropic efficiency of the compressor following the same pattern as the one we collect historical data from.

We do not perform time series analysis, and we aim at selecting sufficient features so that isentropic efficiency can fully depend on them because previous data can be outdated. Given the feature variable, we can assume that efficiencies at different time points are independent.

3.4 Model

3.4.1 Traditional Regression Models

A common method for fault detection is classification. It generates the fault detection or diagnosis results directly. However, the regression method has some advantages over classification. Firstly, since we only calculate what the efficiency should have been when the system follows the old pattern and is not faulty, we do not need faulty data during the training process, and the problem of unbalanced data of normal and faulty ones does not occur. The second advantage is flexibility. Engineers can decide the threshold for deviation according to actual situations, and they can even give different levels of alarm to reflect different levels of severity.

We first test the linear model and SVM regression. Linear regression has already been applied by manufacturers but proved to be not so accurate [33].

We use a similar linear model

$$f(\boldsymbol{x}) = \sum_{i=1}^{D}\sum_{j=1}^{D} a_{ij}x_i x_j + \sum_{k=1}^{D} a_k x_k + a_0, \tag{3.1}$$

where a d-dimension data record $\boldsymbol{x} = [x_1, \ldots, x_D]^T$. Results of linear model and SVM regression are also included in Section 4.2.

3.4.2 Neural Network

Isentropic efficiency of compressors can be affected by many factors, like the working environment, installation, and configuration, so it is difficult to design an accurate physics-based model. Fortunately, we can obtain sufficient data to train a "black-box" model.

The model of neural network can mimic nonlinear relationships, making it a suitable choice. We investigate the influence of the size of the network, batch normalization layers, activation functions, dropout layers and different optimization methods, including MBGD with momentum and Adam algorithm.

3.4.3 Hyperparameter Tuning

Hyperparameters should be set before the training process that generates other parameters.

For linear and SVM regression, we mainly use default values in Python library sklearn.

For neural networks, learning rate and momentum are set mainly according to demos in PyTorch. They mostly affect the convergence during training, and we hope that they can both save time and maintain performance. We test different network structures, including the size of hidden layers, dropout, and batch normalization, and choose the optimal ones.

3.5 Preprocessing

3.5.1 Stability

When the system is on, it needs time to stabilize, and data before that is not important. In practice, a system can be on and off quite frequently, and we should collect only stable data for training. ClimaCheck Sweden AB marks all the unstable data during collection, which are all discarded.

3.5.2 Probability Distribution Estimation

In this subsection, we only exhibit the distribution based on pressures, and other variables share similar characteristics. Figure 3.2 exhibits the probability distribution estimation of high pressure P_H and low pressure P_L of some compressors. We utilize GMM to imitate its distribution because data from nearby time points are usually very similar and form clusters. The latent variable $\boldsymbol{z}$ in Eq. 2.4 represents which cluster the data belongs to. The GMM centers are also exhibited in the plot.

3.5.3 Imbalanced Clustering

The collected data can be imbalanced because the compressor can operate under some pressures for a longer time. If we analyse them with GMM, each data record corresponds to a center of the cluster. If the number of data records in a cluster exceeds a threshold, extra data are randomly discarded. In this way, we can both make the dataset more balanced and limit the size of the dataset, which can reduce the computation resources required.

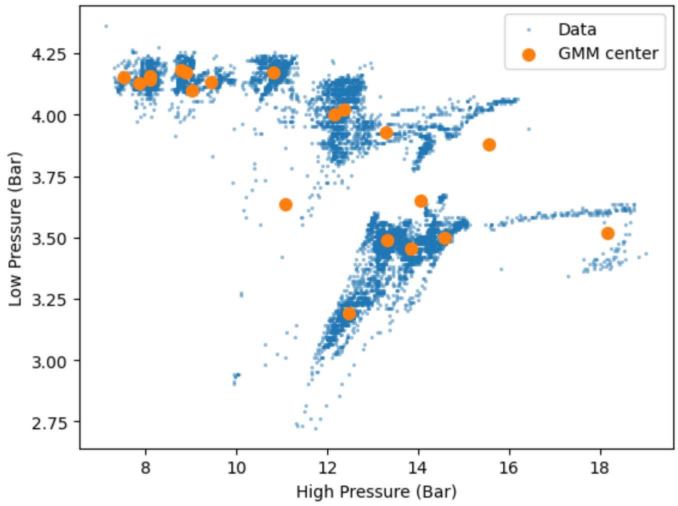
3.5.4 Normalization

The regularization parameter of the linear model and RBF kernel of SVM assume that all features are centered around zero [39] or have variance in the same order.

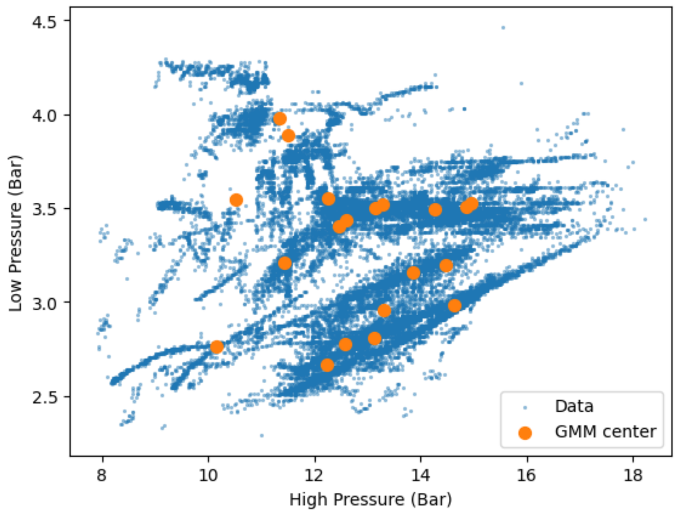
The gradient of sigmoid function $\sigma'(x)$ is close to zero when x is too large or too small, and this can lead to vanishing gradient problem. So it is better if data are centered around 0. Moreover, exploding gradient problem can happen when large gradients are multiplied according to the chain rule.

As a result, the scale of data should be redesigned during preprocessing. There are two possible preprocessing schemes.

The first scheme is to make data of each feature zero-mean and their



(a) System A in September



(b) System B from August 2023 to March 2024

Figure 3.2: Scatter Plot of High and Low Pressures and GMM Centers in Probability Distribution Estimation

variance one:

$$\boldsymbol{x}_t = [x_{t1}, \dots, x_{td}, \dots, x_{tD}], t = 1, \dots, T,$$
$$\hat{\mu}(d) = \sum_{t=1}^{T} x_{td},$$
$$\hat{\sigma}(d) = \sqrt{\frac{\sum_{t=1}^{T}(x_{td} - \hat{\mu}(d))^2}{T}},$$
$$x'_{td} = \frac{x_{td} - \hat{\mu}(d)}{\hat{\sigma}(d)}. \tag{3.2}$$

The empirical mean and standard deviation of data are stored so that they can be used to reconstruct the original data later.

Another possible scheme is to scale the data into $[0, 1]$ as in [3]

$$\boldsymbol{x}_t = [x_{t1}, \dots, x_{td}, \dots, x_{tD}], t = 1, \dots, T,$$
$$max(d) = \max_{t=1,\dots,T} \quad x_{td},$$
$$min(d) = \min_{t=1,\dots,T} \quad x_{td},$$
$$x'_{td} = \frac{x_{td} - min(d)}{max(d) - min(d)}. \tag{3.3}$$

We choose the first scheme because it takes all the data into consideration and is less influenced by outliers.

3.6 Anomaly Detection

If the prediction result deviates too much from the actual one, it should be viewed as an error. The deviation can originate from two sources:

1. The system follows a new (probably faulty) pattern, while the model still imitates the old one.

2. The fault in the system has affected the measurement of isentropic efficiency, while the inputs of the model are unaffected variables.

Also note that there can be short-term deviations in data of the normal period, so single or inconsistent errors should be dismissed to decrease the false alarm rate. Sometimes we can also merely present prediction results as a reference and let engineers decide what to do, which exhibits the flexibility of the method.

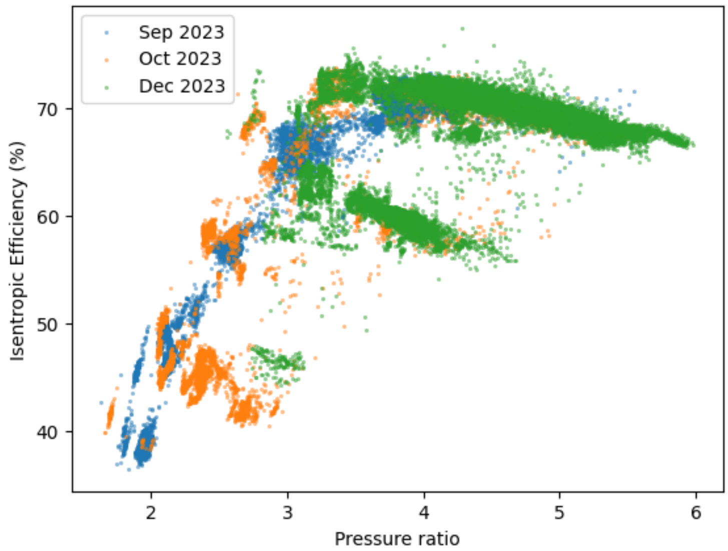


Figure 3.3: Scatter Plot of Isentropic Efficiency and Pressure Ratio in September, October, and December

3.7 Model Update

We decide whether to update the model based on change of data and prediction accuracy.

3.7.1 Probability of Following the Same Distribution

Although we have data from various systems, we still need to consider the problem of insufficient data for new systems. We can train an initial model with data collected during the first period. During the next period, some data can still be similar to the old ones but not for the rest, because the distribution of data can change over time. For example, Figure 3.3 exhibits how data change from September to December. Generally, the range of pressure ratio moves towards higher values. So if we only have data in September for training, the performance can be very poor for October and later.

Empirically, we assume that if new data and training data are from the same distribution, the prediction accuracy will be high. If not, the performance depends on the generalization of the model and cannot be guaranteed. We assume that the training data are random variables with probability density function (pdf) $p_A(\boldsymbol{x})$. The pdf $p_A(\boldsymbol{x})$ can be estimated with observations $A =$

$\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. The new random variable is generated from the pdf $p(\boldsymbol{x})$, and is observed as $\boldsymbol{x}_t$. In order to determine whether a data record can be considered as originating from the distribution of the training dataset, which is $p(\boldsymbol{x}) = p_A(\boldsymbol{x})$, we should check the following relation as an indicator:

$$P(p(\boldsymbol{x}) = p_A(\boldsymbol{x}) | \boldsymbol{x} = \boldsymbol{x}_t) = \frac{p_A(\boldsymbol{x} = \boldsymbol{x}_t) \cdot P(p(\boldsymbol{x}) = p_A(\boldsymbol{x}))}{p(\boldsymbol{x} = \boldsymbol{x}_t)}, \tag{3.4}$$

where $P(p(\boldsymbol{x}) = p_A(\boldsymbol{x}))$ is prior probability. For example, if there is another possible choice $p_B(\boldsymbol{x})$, and we need to decide whether $\boldsymbol{x}_t$ is from $p_A(\boldsymbol{x})$ or $p_B(\boldsymbol{x})$, we need to consider:

$$\frac{P(p(\boldsymbol{x}) = p_A(\boldsymbol{x}) | \boldsymbol{x} = \boldsymbol{x}_t)}{P(p(\boldsymbol{x}) = p_B(\boldsymbol{x}) | \boldsymbol{x} = \boldsymbol{x}_t)} = \frac{p_A(\boldsymbol{x} = \boldsymbol{x}_t) \cdot P(p(\boldsymbol{x}) = p_A(\boldsymbol{x}))}{p_B(\boldsymbol{x} = \boldsymbol{x}_t) \cdot P(p(\boldsymbol{x}) = p_B(\boldsymbol{x}))} \propto \frac{p_A(\boldsymbol{x} = \boldsymbol{x}_t)}{p_B(\boldsymbol{x} = \boldsymbol{x}_t)}. \tag{3.5}$$

This gives rise to Maximum A Posteriori (MAP) or Maximum Likelihood (ML) estimation.

However, in our problem formulation, $p_B(\boldsymbol{x})$ is implicit, so we only check $p_A(\boldsymbol{x} = \boldsymbol{x}_t)$. To compare with a universal threshold, we use its ratio with the highest possible value and use a logarithm. Assume we also use GMM and the component that input $\boldsymbol{x}$ is classified into is the k-th, then the score of the logarithm likelihood ratio we use is

$$\begin{aligned} score(\boldsymbol{x}_t) &= \ln \frac{p_A(\boldsymbol{x} = \boldsymbol{x}_t)}{\max_{\boldsymbol{x}} p_A(\boldsymbol{x})} \\ &= \ln \frac{\frac{1}{\sqrt{(2\pi)^k \boldsymbol{\Sigma}_k}} e^{-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_t - \boldsymbol{\mu}_k)}}{\frac{1}{\sqrt{(2\pi)^k \boldsymbol{\Sigma}_k}}} \\ &= (\boldsymbol{x}_t - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_t - \boldsymbol{\mu}_k). \end{aligned} \tag{3.6}$$

π_k in Eq. 2.2 represents the frequency of data classified into the k-th component. Since this is not considered to be relevant to the score, we use $p(\boldsymbol{x}|k)$ as $p(\boldsymbol{x})$. If the score exceeds the threshold, we can assume that the prediction should be accurate. the threshold is set by experience, which will be elaborated on in Section 4.4.

DBSCAN can perform such tasks as well. Data of a cluster in DBSCAN correspond to $p(\boldsymbol{x}) = p_A(\boldsymbol{x})$, and noisy data in DBSCAN correspond to $p(\boldsymbol{x}) \neq p_A(\boldsymbol{x})$. However, we choose the model of GMM because it has better performance, which will be discussed in Section 4.7.

3.7.2 Dataset Update and Training

In the previous subsection, we have discussed how to decide whether a new data record can be viewed as being generated from the same distribution as the training dataset. For those following the same distribution, we can foresee that the prediction of isentropic efficiency should be accurate unless the pattern of the system has changed, which implies that an alarm should be generated and the data cannot be used for update. Also, if a proportion of new data follows the same distribution and their prediction is accurate, we can know that the system is normal for the corresponding time period. If we further assume that the system is also normal for the rest time period, we can use the new data for update. Considering that the mean and variance of new data can be different, we restart from the preprocessing process, and the old parameters can be set as the initialization parameters of the latest model.

Chapter 4

Results and Analysis

4.1 Datasets

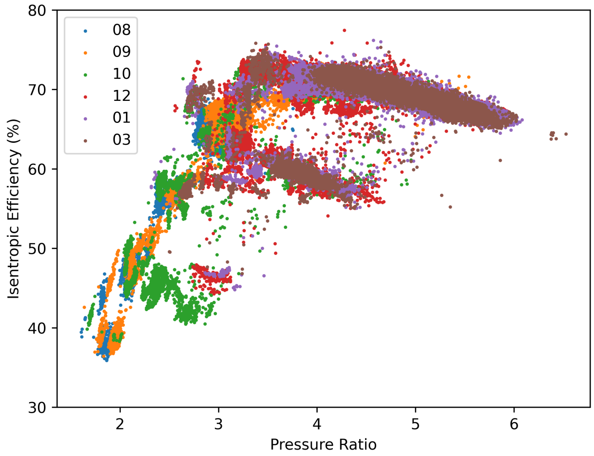
The results are mostly generated from three different systems: System A, B, and C. System A is the most complicated system because it is composed of two different compressors and all the measurements are the means of them for economic reasons. System B is in the same location as A but with a different data structure. System C is one normal subsystem from seven identical subsystems, among which one known subsystem is faulty.

The isentropic efficiency and pressure ratio of systems A, B, and C are shown in Figure 4.1, and they contain 115002, 63205, and 22981 data records accordingly. The data are collected from August 2023 to March 2024, and the numbers on the labels represent these months.

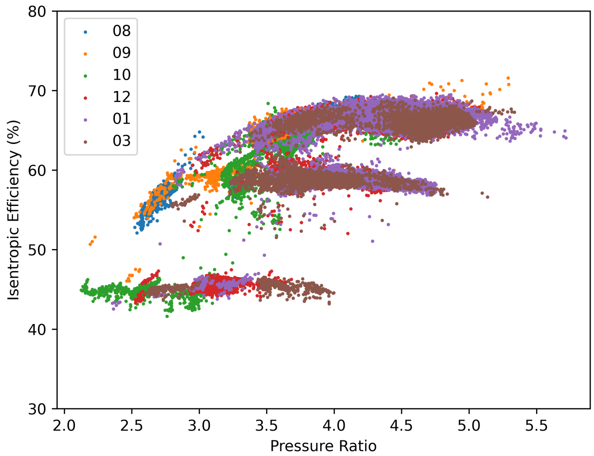
More systems are also tested with the proposed method, but the three systems above are considered to be the most representative ones. Two factors are taken into consideration: diversity of data and generalization of systems. Data from system A change over time and the change is not drastic enough to eliminate the consistency of data. The model and parameters of system B can be used on another similar but non-identical system, which can exhibit the generalization of the model. The main reason to choose C is that its identical subsystems can be used for tests in Section 4.5.

4.2 Offline Training

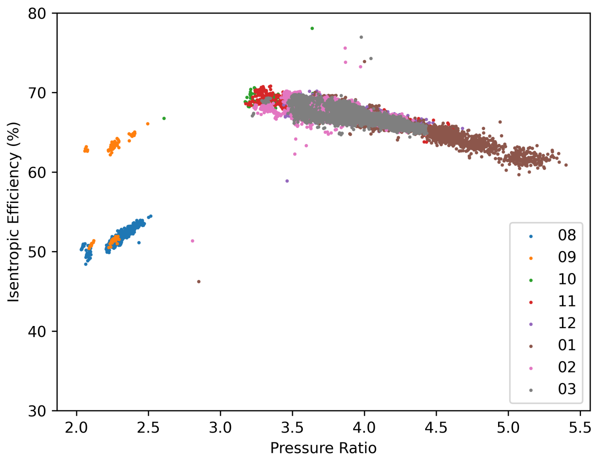
The dataset of each system is divided into the training set, validation set, and test set. We use neural network with fully connected layer and Adam optimizer with a learning rate of 0.001. Training sets are divided into batches with size



(a) Dataset of System A



(b) Dataset of System B



(c) Dataset of System C

Figure 4.1: Scatter Plot of Isentropic Efficiency and Pressure Ratio from August 2023 to March 2024

Table 4.1: Root MSE for Validation Sets

	System		
Network Structure	**A**	**B**	**C**
LR	0.385	0.849	0.649
SVR	0.230	0.648	0.644
(8, 4, 1)	0.324	0.575	0.579
(16, 8, 1)	0.142	0.580	0.575
(32, 16,1)	0.092	0.555	0.556
(16, 8, 1), BN	0.516	0.620	0.750
(16, 8, 1), BN, DO	3.475	1.734	1.562
(16, 8, 1), B	0.231	0.630	0.615
(16, 8, 1), SGD	0.159	0.665	0.635
(16, 8, 1), SI	0.139	0.803	0.606

of 64. Table 4.1 shows the Root MSE for validation sets.

$$\text{Root MSE}(B, f) = \sqrt{\frac{\sum_{t=1}^{T}(y_t - f(\boldsymbol{x}_t))^2}{T}} \tag{4.1}$$

The numbers in parenthesis represent the size of the hidden layer and output layer. BN and DO stand for batch normalization and dropout. B means balanced dataset from the view of GMM. SGD denotes that Adam optimizer is replaced by SGD optimizer with a learning rate of 0.01 and momentum of 0.9, and SI denotes that ReLU activation function is replaced by the sigmoid function.

Besides neural networks, LR denotes the linear regression model in Section 4.2 with $\lambda = 1$. SVR is short for support vector regression, and the hyperparameters are all default values, which are $C = 1$, $\epsilon = 0.1$, and the kernel is RBF with $\gamma = \frac{1}{\#\ of\ features}$. The results of linear regression and support vector regression will be discussed in Section 4.7.

Linear regression has worse performance than neural networks. This is consistent with the fact that manufacturers also apply similar linear models but they exhibit low accuracy. For datasets of large sizes, the computation of SVM can be very slow without preprocessing. The results of SVM regression are slightly worse than neural networks but still acceptable.

The middle rows in Table 4.1 compare the performance of different sizes of hidden layers. We aim at achieving a trade-off between cost and performance, and (16, 8, 1) structure is selected. For system A, reduction to (8, 4, 1)

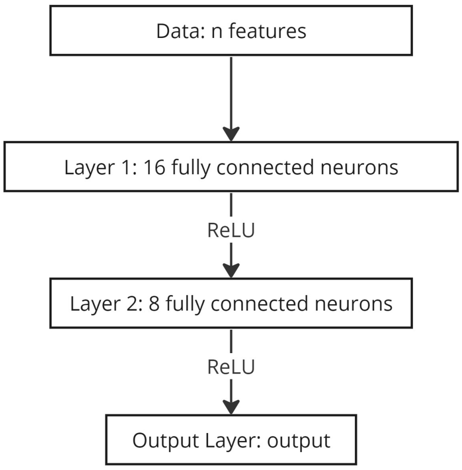


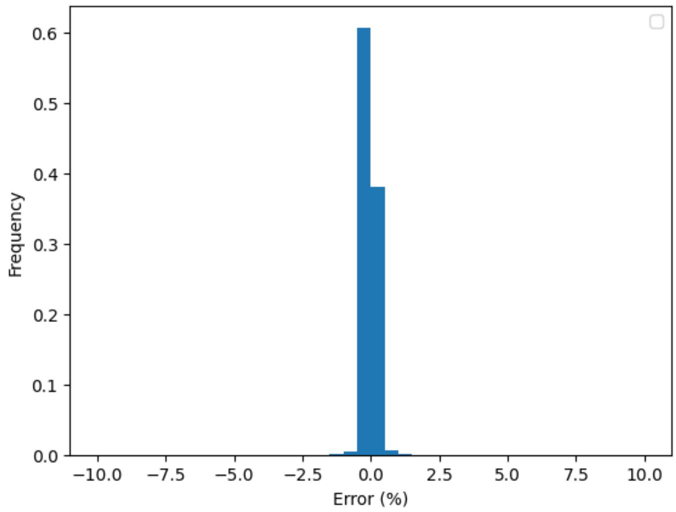
Figure 4.2: Structure of the Neural Network

decreases the performance a lot, probably because the system is more complex than the others, and a larger neural network is required. Moreover, we can see that batch normalization does not help, since we have partly performed similar actions in preprocessing. Dropout can worsen the performance, and maybe it is because the dimension of our method is much lower than image processing methods, where dropout is proved to be useful [45]. Consequently, disabling one dimension of input during training can be too costly. If we use a balanced dataset to train, the performance slightly decreases because the validation set is imbalanced. This shows that the problem of imbalanced data in the systems is not severe. We also try SGD optimizer, and its performance is slightly worse than Adam. The performance of sigmoid and ReLU functions are similar, but training with ReLU has a faster convergence rate.

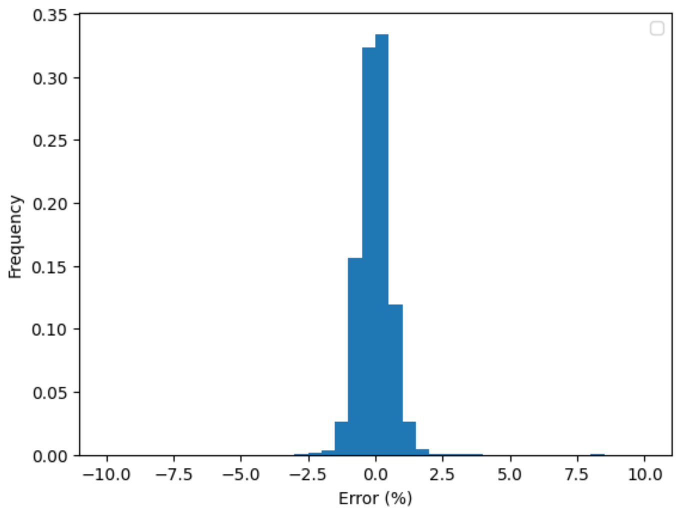
As an example, the (16, 8, 1) structure is chosen and visualized in Figure 4.2. The test results are given through distribution of errors. The columns of Figure 4.3 represent frequencies, so the coloured areas add up to 1. Most of the errors are below 1 percent.

4.3 Log Likelihood Ratio and Prediction Accuracy

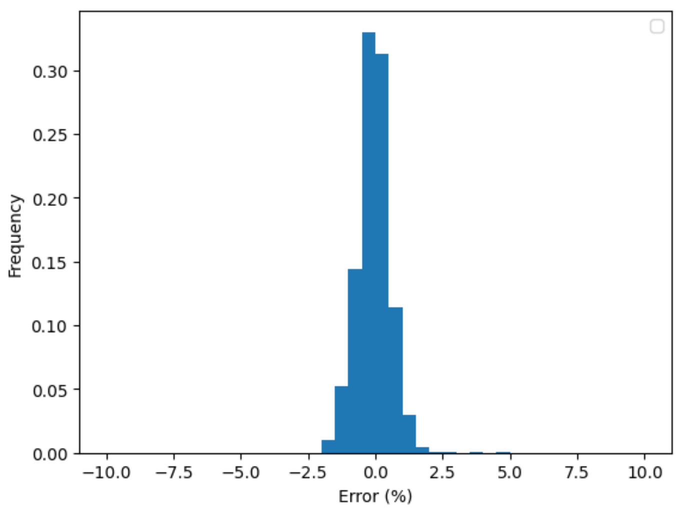
Figure 4.4 exhibits some data of system A in October and corresponding prediction with the model trained from data in September. Empirically, we



(a) Error of System A (Root MSE: 0.144)



(b) Error of System B (Root MSE: 0.588)



(c) Error of System C (Root MSE: 0.579)

Figure 4.3: Histogram of Test Errors

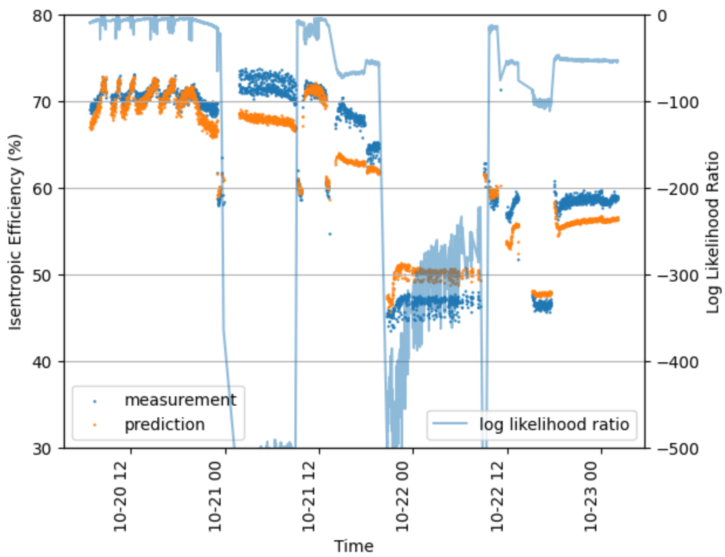


Figure 4.4: Predictions Compared with Measurements and Logarithm Likelihood Ratio from October 20 to 23

can find out that if the new data record is more probable to come from the distribution of the training set, its prediction is more accurate.

Next, we plot all the new data records by their scores of logarithm likelihood ratio in Eq. 3.6 and their prediction error in Figure 4.5. If the score is higher than -50, the errors of data are mainly less than 3 percent. And this can not be guaranteed otherwise. Some predictions are still accurate when the score is low, and this is related to the generalization characteristic of the model.

4.4 Online Update

In Section 4.3, we have explored the relation of prediction error and score of logarithm likelihood ratio with system A, and set the threshold of score at -50. We then assess the prediction of training data in September and test data in October and find out that 40.5% of test data follow the distribution of training data and 97.1% of the 40.5% achieve error lower than 3%. As a result, we can update the model with data in October. Figure 4.6 exhibits the difference in prediction with and without the model update, and the updated model can better predict during some time intervals.

In this case, an update in October is critical for prediction of data from

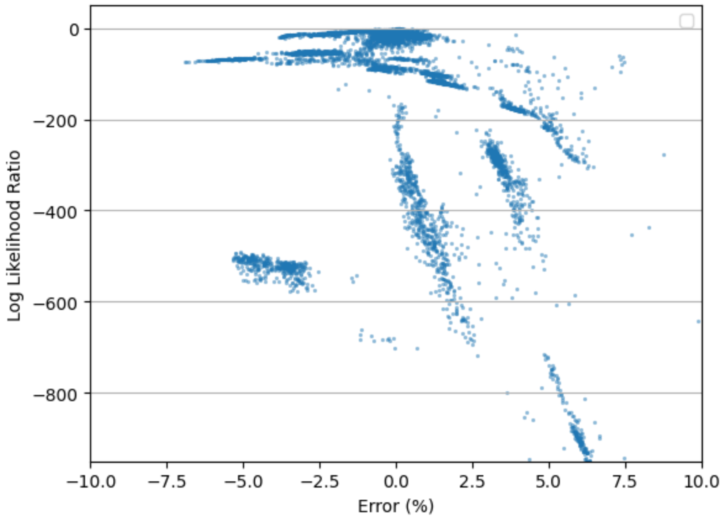


Figure 4.5: Relation between Error and Logarithm Likelihood Ratio

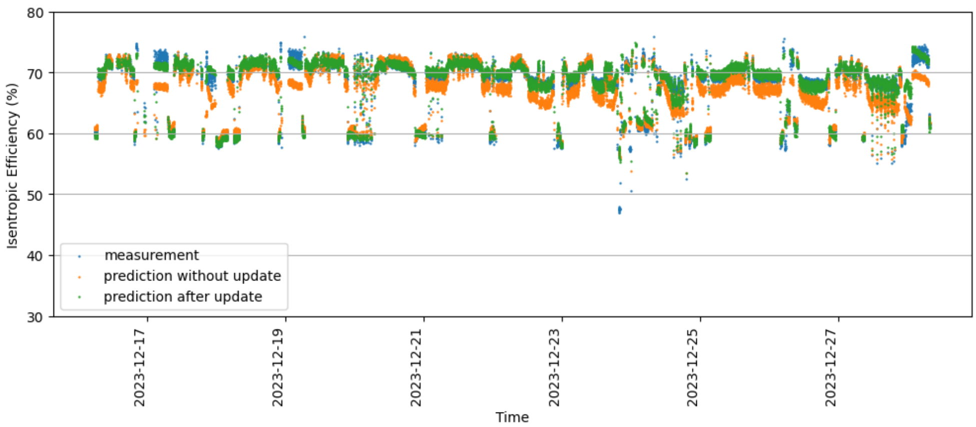


Figure 4.6: Measurements and Predictions with and without Update

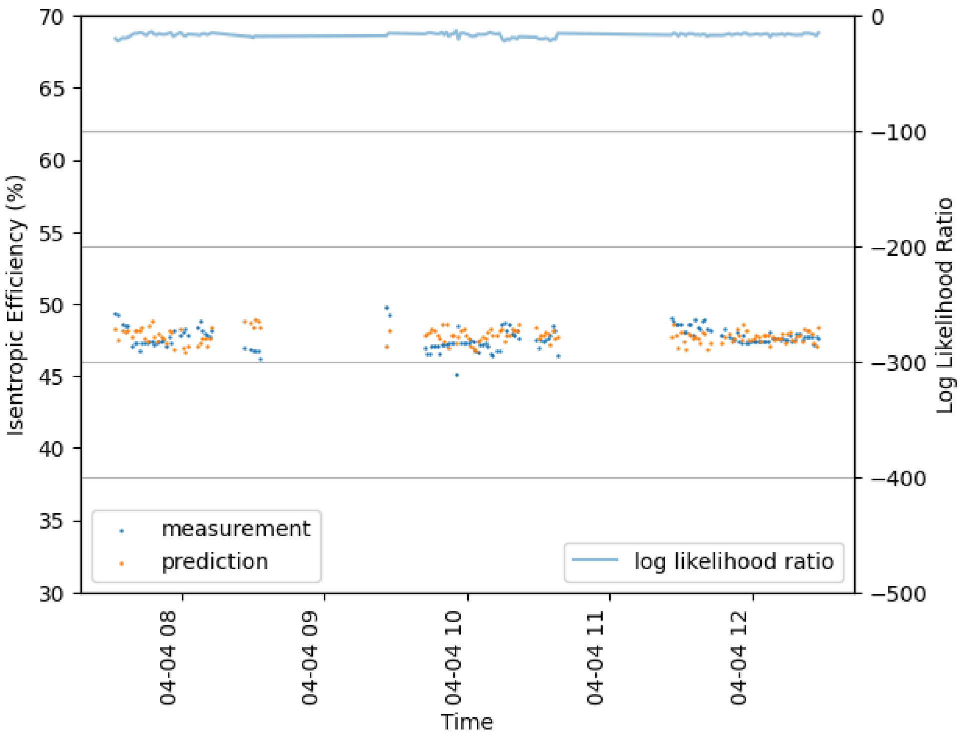


Figure 4.7: Measurements and Predictions of a New System with Model Trained from System B

October to December, while an update in December is not that significant. This is because the change of data in October is more drastic than afterwards.

4.5 Generalization to Similar Systems

We also test whether the model and parameters can be applied to a similar system.

There is a system similar to system B. It was recently installed and does not have sufficient data for a model. We utilized the model of system B to test this new system, and the result is shown in Figure 4.7. The predictions are accurate because the data follow a similar distribution as well.

As is mentioned in Section 4.1, system C has five similar and fault-free systems, and we name them system D to system H. Their design and manufacturing are the same, but installation and working environments can cause the difference. Table 4.2 shows the 1st and last 20-quantiles of prediction

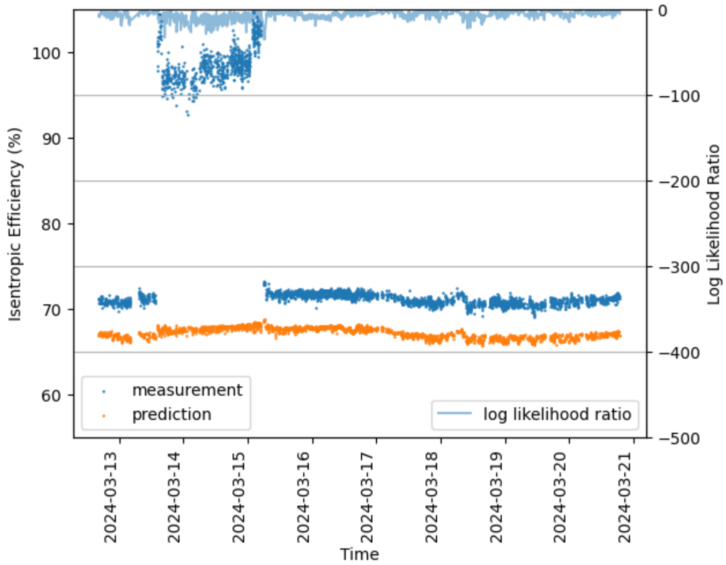


Figure 4.8: Measurements and Predictions of Faulty System

errors among these systems, so the range defined by the quantiles cover 90% of the errors. For example, we use history data from C to train the model, and test with history data from C to H, and this forms the first column of the table. According to the table, we know that the prediction error can be higher if training and test data are from different but similar systems. We can also see the direction of deviation of errors between measured and predicted values, and this is why we use quantiles instead of Root MSE in the table. However, there is no suitable way to validate our results, and they can be used as a reference. Usually, an error within 5 percent is assumed to be acceptable.

4.6 Fault Detection

Besides the fault-free systems, system C also has a similar system with a severe fault. When we use the model from system C and test with the faulty system, the result is exhibited in Figure 4.8 .

There is a refrigerant liquid carry-over issue in the system, and the measurement is affected, so there can be an apparent isentropic efficiency at almost 100 percent due to cooling of the compressor (this affects lubrication and results in a high risk of compressor failure). According to our prediction, the efficiency should have been around 67 percent. Also, we can see that test

Table 4.2: 1st and last 20-Quantiles of Errors between Similar Systems

Test \Train	**C**	**D**	**E**	**F**	**G**	**H**
C	-1.113% ~ 0.832%	-1.898% ~ 0.848%	-2.813% ~ -0.450%	-1.593% ~ 1.229%	-0.091% ~ 1.974%	-0.128% ~ 2.505%
D	-0.906% ~ 1.743%	-1.218% ~ 1.355%	-2.614% ~ 0.167%	-0.876% ~ 2.049%	0.138% ~ 2.909%	0.527% ~ 3.146%
E	-0.020% ~ 2.529%	-0.832% ~ 2.743%	-0.728% ~ 0.784%	-1.306% ~ 2.989%	1.555% ~ 3.577%	2.007% ~ 4.127%
F	-1.107% ~ 2.381%	-1.832% ~ 2.218%	-3.017% ~ 1.268%	-0.760% ~ 0.909%	-0.103% ~ 3.596%	0.295% ~ 4.093%
G	-2.160% ~ -0.567%	-3.249% ~ -0.243%	-3.632% ~ -2.009%	-3.705% ~ -0.086%	-0.598% ~ 0.578%	-0.635% ~ 1.496%
H	-3.032% ~ -0.759%	-4.123% ~ -0.964%	-3.870% ~ -2.381%	-4.199% ~ -0.365%	-0.964% ~ 0.677%	-0.670% ~ 0.655%

data has a high score of logarithm likelihood ratio, so the prediction is reliable. As a consequence, the continuous abnormal measurement should be viewed as an error.

4.7 Discussion

As is mentioned in Section 3.7.1, we can use DBSCAN to replace GMM, and a cluster in DBSCAN corresponds to a component in GMM. However, in tests, we find out that many data with accurate prediction are categorized as noisy data, and data in clusters can have poor accuracy. The problem is not solved when we tune the hyperparameters. A possible explanation is that this algorithm relies too much on margin data, while the model is trained with the whole dataset. Moreover, we choose GMM over other clustering methods like K-means or DBSCAN, because it takes variance into consideration and can return probability density, which grants us more flexibility in the following steps like Section 4.4.

Training of neural networks is composed of many epochs till convergence. One epoch implies that all the data in the dataset are trained for a round. An empirical finding is that models trained with more epochs and closer to convergence have better characteristics of generalization. However, when it comes to updating models, one or two epochs can be sufficient. Maybe it is because the model has been well-trained and just needs to be adjusted. In contrast, old parameters of linear and SVM regression models can not be utilized during update.

Due to randomness in the training process, the results can be slightly different when replicating. However, the absolute difference between different times of training is usually less than 0.5 in root MSE.

Besides the factors we select, the system itself can also decide the scale of error. This might be because some systems are more noisy and data are affected by differences in sensors as well.

We do not use synthetic data to test because we are not very clear about the principles of generating synthetic data, and tests on real data are more reliable.

The results in this chapter focus on the condition monitoring of isentropic efficiency and are proven to be effective. To perform the automated fault detection, the only task left is to compare the difference between predictions and measurements with a threshold. The engineers can set thresholds by experience, so it is not highlighted here.

Chapter 5

Conclusions and Future work

5.1 Conclusions

Isentropic efficiency of the compressor can serve as an indicator of operating conditions of heat pump and refrigeration systems, so we perform predictive maintenance through the prediction of isentropic efficiency. We choose features that are unaffected by faults in compressors for the model through expert knowledge, and can draw the following conclusions about the neural network model:

1. Neural networks outperform traditional regression methods like linear regression and support vector regression.
2. Fully connected layers of medium size are sufficient, as a trade-off between cost and performance.
3. Activation function of ReLU function outperforms sigmoid function from the perspective of convergence speed.
4. Performance of Adam optimizer slightly exceeds SGD.
5. Dropout and batch normalization cannot enhance the performance.

Data of selected input features can change over time, and we can utilize GMM to model the distribution of the dataset. If we calculate the probability that new data records originate from the same distribution as the training dataset, we can foresee the prediction accuracy. If the data record matches the probability distribution of the training set, the prediction should be accurate unless the system is operating under another pattern that implies faults in it.

If a proportion of new data follow the distribution of the training set, and their predictions are accurate, we can assume that the heat pump and refrigeration system is fault-free and the latest data can be used for updating the dataset and retraining.

If we use the model of a system on a similar system, the prediction result is influenced by an additional bias, which is unpredictable and decided by the unique conditions of the system itself. Thus, we lack a means of evaluating such results, and they can just work as a reference for engineers to ensure the system is not suffering from a severe fault, assuming that our results are correct.

5.2 Limitations

The limitations of the thesis mainly come from the generalization of the model, including the aspect of new data and similar systems.

Since there are new data that follow a new probability distribution, we need to update the model on time. During the update of the model, we assume that the system is fault-free when we can ensure this for a proportion of time period and cannot make valid assessments for the rest. This assumption may not hold all the time.

From the results in Section 4.4, data with a higher score of logarithm likelihood ratio have lower errors, but some of the other data also have low errors. There might be a better method to distinguish between data with high and low errors.

When updating the model, the diversity of the data is enhanced but the system itself remains the same, so we can still evaluate our method through deviations between predictions and measurements. However, if we apply the model to another similar system, we can not expect that the prediction can still be accurate. In practical engineering, most compressors do not work as they are designed to, due to installation problems and other factors. So they do not work under the best mode, but still generally acceptable. As a result, we can basically only detect severe faults from large deviations, otherwise there will be too many false alarms if we categorize them as false alarms. Usually, the deviations are moderate and consistent, and we can only say that these results are in accordance with our expectations, but cannot evaluate them.

5.3 Future work

We have tested our method with different types of heat pump and refrigeration systems, and generally, the more we know about the systems, the more accurate our predictions can be. And for systems we do not know much about, we can foresee that our prediction performance will not be very high. However, is it possible to calculate the upper bound of accuracy given the fixed information, like the calculation of channel capacity in information theory? Of course, a "black-box" model is not sufficient for this and more details of the systems and data should be studied.

We can also try to differentiate the rare minor fault and common sub-optimal working conditions discussed in Section 5.2, which can all be exhibited in moderate deviations.

Currently, the research object is only compressors, but maybe the methodology can be extended to other parts of heat pump and refrigeration systems, which can provide more detailed maintenance for the whole system.

We have chosen stable data for training and testing, but there still exist very few noisy data records, although this does not affect the result in general. We can try denoising or modifying the algorithm that selects the stable data.

We do not investigate how the level of noise affects the proposed method, because we are not sure about what the noise in the systems is like.

We use GMM for probability distribution estimation of data, and we can further study this for better models.

The work of the thesis can be viewed as part of a digital twin [50], so we can also study how to design a digital twin for the systems.

5.4 Ethics and Sustainability

Data science requires a large number of data to train the models, and the quality and quantity of the data used to train the models can significantly affect the performance. If the collected data are not handled in a responsible way, privacy leakage can occur and the interest of customers is threatened.

Equality between different regions should also be taken into consideration. In less developed regions, there may not be enough sensors monitoring the systems and publishing data through the Internet. As a result, the quality and quantity of data from developing countries may be reduced, and a lack of representativeness may lead to lower performance for customers in developing countries.

In fault detection, a threshold should be specified, which involves a trade-off between cost and performance. We may have to adjust the threshold depending on different circumstances. For example, we should attempt to detect all the potential faults in systems in hospitals, even though this means that more resources will be put into use. Of course, the decision-making process should be disclosed to parties affected by it.

5.5 Social Impact

The thesis has the following significance for the society:

1. The project monitors the conditions of heat pump and refrigeration systems and detects faults. Earlier detection of faults can enormously reduce energy waste and downtime.

2. The proposed method considers the change of pattern in data, making the model suitable for practical cases.

3. The project uses and analyses actual data from the systems, making it easier for commercial use.

4. The users can customize the fault detection task depending on the desirable false alarm rate.

5. The work provides useful information for further research, like fault diagnosis, digital twin, etc.

References

[1] M. Elnour and N. Meskin, “Novel actuator fault diagnosis framework for multizone hvac systems using 2-d convolutional neural networks,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1985–1996, 2022. doi: 10.1109/TASE.2021.3067866 [Pages 1, 7, and 8.]

[2] B. Sun, P. B. Luh, Q.-S. Jia, Z. O’Neill, and F. Song, “Building energy doctors: An spc and kalman filter-based method for system-level fault detection in hvac systems,” *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 215–229, 2014. doi: 10.1109/TASE.2012.2226155 [Pages 3, 6, and 7.]

[3] H. Nguyen, K. Tran, S. Thomassey, and M. Hamad, “Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management,” *International Journal of Information Management*, vol. 57, p. 102282, 2021. doi: https://doi.org/10.1016/j.ijinfomgt.2020.102282. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026840122031481X [Pages 5 and 26.]

[4] A. Alzghoul, B. Backe, M. Löfstrand, A. Byström, and B. Liljedahl, “Comparing a knowledge-based and a data-driven method in querying data streams for system fault detection: A hydraulic drive system application,” *Computers in Industry*, vol. 65, no. 8, pp. 1126–1135, 2014. doi: https://doi.org/10.1016/j.compind.2014.06.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361514001250 [Page 5.]

[5] K.-H. Lai, L. Wang, H. Chen, K. Zhou, F. Wang, H. Yang, and X. Hu, “Context-aware domain adaptation for time series anomaly detection,” 2023. [Page 6.]

[6] A. Alghanmi, A. Yunusa-Kaltungo, and R. Edwards, "A comparative study of faults detection techniques on hvac systems," in *2021 IEEE PES/IAS PowerAfrica*, 2021. doi: 10.1109/PowerAfrica52236.2021.9543158 pp. 1–5. [Pages 7 and 9.]

[7] R. van de Sand, C. Falk, S. Corasaniti, and J. Reiff-Stephan, "A data-driven fault diagnosis approach towards oil retention in vapour compression refrigeration systems," in *2019 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)*, 2019. doi: 10.1109/CANDO-EPE47959.2019.9111046 pp. 197–202. [Pages 7 and 8.]

[8] F. Simmini, M. Rampazzo, F. Peterle, G. A. Susto, and A. Beghi, "A self-tuning kpca-based approach to fault detection in chiller systems," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 4, pp. 1359–1374, 2022. doi: 10.1109/TCST.2021.3107200 [Pages 7 and 9.]

[9] C. Guo, Y. Wang, and Y. Zhong, "Air-conditioning fault prediction about new energy bus based on particle swarm algorithm," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2017. doi: 10.1109/IAEAC.2017.8054008 pp. 211–215. [Pages 7 and 8.]

[10] M. Meyer, M. Gergeleit, and D. Krechel, "Development of anomaly detection with variable contexts on refrigeration data," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2023. doi: 10.1109/ETFA54631.2023.10275472 pp. 1–4. [Pages 7 and 9.]

[11] S. Hatanaka and H. Nishi, "Efficient gan - based unsupervised anomaly sound detection for refrigeration units," in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 2021. doi: 10.1109/ISIE45552.2021.9576445 pp. 1–7. [Pages 7 and 9.]

[12] S. Gupta and B. Gupta, "Energy anomaly detection and modelling on smart premises using sdar," in *2021 International Conference on Industrial Electronics Research and Applications (ICIERA)*, 2021. doi: 10.1109/ICIERA53202.2021.9726730 pp. 1–5. [Pages 7 and 10.]

[13] T. S. Anttal, "Smart home dataset with weather information," 2019. doi: https://www.kaggle.com/datasets/taranvee/smart-home-dataset-with-weather-information [Pages 7 and 10.]

[14] X. Wang and K. Yan, “Fault detection and diagnosis of hvac system based on federated learning,” in *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2022. doi: 10.1109/DASC/PiCom/CBDCom/Cy55231.2022.9927944 pp. 1–8. [Pages 7 and 8.]

[15] F. Xiao, C. Zheng, and S. Wang, “A fault detection and diagnosis strategy with enhanced sensitivity for centrifugal chillers,” *Applied Thermal Engineering*, vol. 31, no. 17, pp. 3963–3970, 2011. doi: https://doi.org/10.1016/j.applthermaleng.2011.07.047 SET 2010 Special Issue. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359431111004078 [Page 7.]

[16] J. Wall, Y. Guo, J. Li, and S. West, “A dynamic machine learning-based technique for automated fault detection in hvac systems,” vol. 117, 01 2011. [Page 7.]

[17] A. Bashi, V. P. Jilkov, and X. R. Li, “Fault detection for systems with multiple unknown modes and similar units and its application to hvac,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 957–968, 2011. doi: 10.1109/TCST.2010.2062183 [Pages 6 and 7.]

[18] Z. Soltani, K. K. Soerensen, J. Leth, and J. D. Bendtsen, “Fault detection of supermarket refrigeration systems using convolutional neural network,” in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020. doi: 10.1109/IECON43393.2020.9254485 pp. 231–238. [Pages 7 and 8.]

[19] Z. Soltani, K. Kjaer Soerensen, J. Leth, and J. Dimon Bendtsen, “Robustness analysis of pca-svm model used for fault detection in supermarket refrigeration systems,” in *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2021. doi: 10.1109/ICECCE52056.2021.9514086 pp. 1–6. [Pages 7 and 8.]

[20] Y. Liu, X. Hei, J. Zhao, Y. Zhang, and G. Xie, “Fault diagnosis based on prior knowledge for train air-conditioning unit,” in *2015 11th International Conference on Computational Intelligence and Security (CIS)*, 2015. doi: 10.1109/CIS.2015.24 pp. 66–69. [Pages 7 and 8.]

[21] S. Takeuchi and T. Saito, “Fault diagnosis method based on scaling law for on-line refrigerant leak detection,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018. doi: 10.1109/ICMLA.2018.00177 pp. 1087–1094. [Pages 6 and 7.]

[22] Y. Zhao, F. Xiao, and S. Wang, “An intelligent chiller fault detection and diagnosis methodology using bayesian belief network,” *Energy and Buildings*, vol. 57, pp. 278–288, 2013. doi: https://doi.org/10.1016/j.enbuild.2012.11.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778812005968 [Pages 7 and 8.]

[23] D. A. M. Lemes, T. W. Cabral, L. L. Motta, G. Fraidenraich, E. R. de Lima, F. B. Neto, and L. G. P. Meloni, “Low runtime approach for fault detection for refrigeration systems in smart homes using wavelet transform,” *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2023. doi: 10.1109/TCE.2023.3328147 [Pages 7 and 10.]

[24] T. Yokouchi and M. Kondo, “Lstm-based anomaly detection for railway vehicle air-conditioning unit using monitoring data,” in *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, 2021. doi: 10.1109/IECON48115.2021.9589928 pp. 1–6. [Pages 7 and 9.]

[25] Y. Yan, P. B. Luh, and K. R. Pattipati, “Fault diagnosis of hvac air-handling systems considering fault propagation impacts among components,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 705–717, 2017. doi: 10.1109/TASE.2017.2669892 [Pages 6 and 7.]

[26] L. Wang, M. Turowski, M. Zhang, T. Riedel, M. Beigl, R. Mikut, and V. Hagenmeyer, “Point and contextual anomaly detection in building load profiles of a university campus,” in *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, 2020. doi: 10.1109/ISGT-Europe47291.2020.9248792 pp. 11–15. [Pages 7 and 9.]

[27] Y. Guo, C. Liu, J. Dang, and H. Chen, “Regression method based fault detection of the vrf system for energy saving,” in *2021 IEEE 4th International Electrical and Energy Conference (CIEEC)*, 2021. doi: 10.1109/CIEEC50170.2021.9510804 pp. 1–5. [Pages 7 and 9.]

[28] M. Dey, S. P. Rana, and S. Dudley, "Semi-supervised learning techniques for automated fault detection and diagnosis of hvac systems," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2018. doi: 10.1109/ICTAI.2018.00136 pp. 872–877. [Pages 7 and 8.]

[29] K. Yan, Z. Ji, H. Lu, J. Huang, W. Shen, and Y. Xue, "Fast and accurate classification of time series data using extended elm: Application in fault diagnosis of air handling units," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1349–1356, 2019. doi: 10.1109/TSMC.2017.2691774 [Pages 7 and 8.]

[30] M. Y. L. Chew and K. Yan, "Intelligent and automated fault detection and diagnosis strategy for hvac systems based on maintainability rules for construction 4.0," in *2021 IEEE 23rd Int Conf on High Performance Computing Communications; 7th Int Conf on Data Science Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud Big Data Systems Application (HPCC/DSS/SmartCity/DependSys)*, 2021. doi: 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00291 pp. 1946–1951. [Page 8.]

[31] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient gan-based anomaly detection," 2019. [Page 9.]

[32] A. Beghi, R. Brignoli, L. Cecchinato, G. Menegazzo, M. Rampazzo, and F. Simmini, "Data-driven fault detection and diagnosis for hvac water chillers," *Control Engineering Practice*, vol. 53, pp. 79–91, 2016. doi: https://doi.org/10.1016/j.conengprac.2016.04.018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066116300880 [Page 10.]

[33] "Data-driven compressor performance maps and cost correlations for small-scale heat-pumping applications," *Energy*, vol. 291, p. 130171, 2024. doi: https://doi.org/10.1016/j.energy.2023.130171. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S036054422303565X [Pages 10, 12, and 22.]

[34] M. C. Keir and A. G. Alleyne, "Dynamic modeling, control, and fault detection in vapor compression systems," 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:58205682 [Pages vii and 11.]

[35] "enthalpy," 2019. doi: doi:10.1351/goldbook.E02141. [Online]. Available: https://doi.org/10.1351/goldbook.E02141 [Page 12.]

[36] E. Lemmon, M. Huber, and M. McLinden, "Nist standard reference database 23: Reference fluid thermodynamic and transport properties-refprop, version 9.1," 2013-05-07 2013. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=912382 [Page 12.]

[37] Y. A. Cengel and M. A. Boles., *Thermodynamics: An Engineering Approach. 7th Edition.* New York: Mcgraw-Hill, 2012. [Page 12.]

[38] E. Shungarov and S. Garanov, "Comparison of the characteristics of low-temperature scroll compressors for heat pump applications," *Chemical and Petroleum Engineering*, vol. 56, pp. 378–384, 09 2020. doi: 10.1007/s10556-020-00784-x [Page 12.]

[39] C. Bishop, *Pattern Recognition and Machine Learning*, 01 2006. [Pages 12, 14, 15, 17, 24, and 55.]

[40] E. Wit, E. v. d. Heuvel, and J.-W. Romeijn, "'all models are wrong...': an introduction to model uncertainty," *Statistica Neerlandica*, vol. 66, no. 3, pp. 217–236, 2012. doi: https://doi.org/10.1111/j.1467-9574.2012.00530.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9574.2012.00530.x [Page 13.]

[41] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231. [Page 14.]

[42] F. Pedregosa, 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, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Pages 14 and 15.]

[43] V. Vapnik, *The Nature of Statistical Learning Theory*, 01 2000. ISBN 978-1-4419-3160-3 [Page 15.]

[44] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. doi: 10.1109/5.726791 [Page 16.]

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, pp. 84 – 90, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:195908774 [Pages 16 and 34.]

[46] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” 2012. [Page 16.]

[47] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015. [Page 16.]

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” 2019. [Page 17.]

[49] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017. [Page 17.]

[50] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu, “A comprehensive review of digital twin – part 1: Modeling and twinning enabling technologies,” 2022. [Page 45.]

Appendix A

EM algorithm for GMM

We express GMM in form of Eq. 2.4, and let parameters $\boldsymbol{\theta} = \pi_1, \ldots, \pi_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$, observed dataset $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ and latent variables $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]$, where $\boldsymbol{z}_n = [z_{n1}, \ldots, z_{nK}]$ And EM algorithm [39] consists of the following steps.

1. Initialize the parameters $\boldsymbol{\theta}$.
2. E step. Evaluate $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old})$

$$\gamma(z_{nk}) = p(z_{nk} = 1|\boldsymbol{x}_n) = \frac{\pi_k N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1} K\pi_j N(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{A.1}$$

3. M step. Evaluate

$\boldsymbol{\theta}^{new} = \text{argmax}_{\boldsymbol{\theta}} \quad Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{old}) = \text{argmax}_{\boldsymbol{\theta}} \quad \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old}) p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$

$$\begin{aligned}
\boldsymbol{\mu}_k^{new} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \\
\boldsymbol{\Sigma}_k^{new} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{new})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{new})^T \\
\pi_k^{new} &= \frac{N_k}{N}
\end{aligned} \tag{A.2}$$

where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$

4. Check for convergence. Evaluate $p(\boldsymbol{X}|\boldsymbol{\theta})$, and if the convergence criterion is not met, return to step 2.

TRITA-EECS-EX-2024:306
Stockholm, Sweden 2024

www.kth.se