APPLICATION OF DATA-DRIVEN TECHNIQUES FOR THERMAL MANAGEMENT IN DATA CENTERS

APPLICATION OF DATA-DRIVEN TECHNIQUES FOR THERMAL MANAGEMENT IN DATA CENTERS

BY

KAI JIANG, B.Eng.,M.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Ph.D. of Mechanical Engineering (2021) McMaster University

(Mechanical Engineering) Hamilton, Ontario, Canada

TITLE:	Application of data-driven techniques for thermal management in data centers
AUTHOR:	Kai Jiang M.Eng. (Thermal Engineering) Shanghai Maritime University, Shanghai, China B.Eng. (Nuclear Engineering and Technology) South China University of Technology, Guangzhou, China
SUPERVISOR:	Prof. Fengjun Yan
NUMBER OF PAGES:	xx, 167

Lay Abstract

This thesis mainly investigates the applications of data-driven techniques for thermal management in data centers. The implementations of thermal modeling, temperature estimation and temperature control in data centers are the key contributions in this work. First, we design a data-driven statistical model to describe the complicated thermal dynamics of data center. Then based on the data-driven model, efficient observer and controller are developed respectively to optimize the thermal management in data centers. Moreover, to improve the nonlinear modeling performance in data centers, specific deep input convex neural networks capable of good representation capability and control tractability are adopted. This thesis also proposes two novel strategies to avoid the influence of catastrophic forgetting and noisy data respectively during the training processes. Finally, all the proposed techniques are validated in real experiments or experimental-data-based simulations.

Abstract

This thesis mainly addresses the problems of thermal management in data centers (DCs) through data-driven techniques. For thermal management, a temperature prediction model in the facility is very important, while the thermal modeling based on first principles in DCs is quite difficult due to the complicated air flow and heat transfer. Therefore, we employ multiple data-driven techniques including statistical methods and deep neural networks (DNNs) to represent the thermal dynamics. Then based on such data-driven models, temperature estimation and control are implemented to optimize the thermal management in DCs.

The contributions of this study are summarized in the following four aspects: 1) A data-driven model constructed through multiple linear Autoregression exogenous (ARX) models is adopted to describe the thermal behaviors in DCs. On the basis of such data-driven model, an observer of adaptive Kalman filter is proposed to estimate the temperature distribution in DC. 2) Based on the data-driven model proposed in the first work, a data-driven fault tolerant predictive controller considering different actuator faults is developed to regulate the temperature in DC. 3) To improve the modeling accuracy, a deep input convex neural network (ICNN) is adopted to implement thermal modeling in DCs, which is also specifically designed for further control design. Besides, the algorithm of elastic weight consolidation (EWC) is employed

to overcome the catastrophic forgetting in continual learning. 4) A novel example reweighting algorithm is utilized to enhance the robustness of ICNN against noisy data and avoid overfitting in the training process. Finally, all the proposed approaches are validated in real experiments or experimental-data-based simulations.

To my wife Shuting Zhou,

for her constant love and support.

Acknowledgements

Firstly, I would like to express my great appreciation to my supervisor, Prof. Fengjun Yan, for his patient guidance and constant support throughout my Ph.D. study. He taught me scientific knowledge, research-led approach and the spirit of active exploration and innovation. Under his efficient supervision, I propose creative ideas, conduct effective experiments, and accomplish great achievements. Without his persistent help, the goals of my research would not have been realized.

I also want to thank my advisory committee members, Prof. Ali Emadi and Prof. Prashant Mhaskar, for their excellent insights and constructive suggestions that significantly help to improve the quality of the thesis.

I am deeply grateful to Dr. Souvik Pal and Dr. Suvojit Ghosh for their generous help and guidance in my experiments and project. The experience in Computing Infrastructure Research Centre is a precious treasure in my life. I also want to express my thanks to my lab mates, Dr. Peiying Jennifer Tsai, Dr. Hosein Moazamigoodarzi, Dr. Masoud Kheradmandi, Shizhu Shi, Sahar Asgari and Fernando Martinez Garcia, for their assistance and collaboration in my work.

Finally, I wish to acknowledge the support and great love of my parents and my wife. They keep me going on and this work would not have been possible without their input.

Declaration of Academic Achievements

This dissertation is written to fulfill the requirements of Ph.D. degree in Mechanical Engineering. All the works are accomplished from September 2017 to April 2021. In my Ph.D. program, multiple data-driven modeling methods including statistical approach and DNNs are adopted to represent the thermal dynamics in a DC with rack-based cooling architecture. Based on the data-driven models, efficient observers and controllers are developed to estimate the temperature distribution and regulate the temperature in DCs respectively. Besides, novel ICNNs considering control tractability are also employed to improve the modeling performance. In the ICNNs, specific algorithms are utilized to relief the negative effects of catastrophic forgetting and noisy training data during training processes.

In my Ph.D. study, I have finished 5 manuscripts, and I am the first author of all of them. Two of these papers have been publish in the international journals and the rests are under review. I take initiative to propose the ideas, design the algorithm, conduct the experiments, and collect the data reported in the manuscripts. Dr. Fengjun Yan provides the constructive suggestions and helpful guidance to improve the proposals and implement the simulations and experiments. The papers are listed

as following:

1. Kai Jiang, Shizhu Shi, Hosein Moazanigoodarzi, Chuan Hu, Souvik Pal, Fengjun Yan, Temperature distribution estimation via data-driven model and adaptive Kalman filter in modular data centers, Published in Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2020.

2. Kai Jiang, Masoud Kheradmandi, Chuan Hu, Souvik Pal, and Fengjun Yan, Data-driven fault tolerant predictive control for temperature regulation in data center with rack-based cooling architecture, Under review in Mechatronics, 2021.

3. Kai Jiang, and Fengjun Yan, Thermal Modeling in Data Center via Input Convex Neural Networks overcoming Catastrophic Forgetting, Under review in Future Generation Computer Systems, 2021.

4. Kai Jiang, and Fengjun Yan, Thermal Modeling in Data Centers through Input Convex Neural Networks with Noisy Data, Under review in Neurocomputing, 2021.

5. Kai Jiang, Chuan Hu, and Fengjun Yan, Path-following Control of Autonomous Ground Vehicles based on Input Convex Neural Networks, Published in Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2021.

Contents

List of Figures

List of Tables

Notation, Definitions, and Abbreviations

Abbreviations

ANN	Artificial Neural Network
ARX	AutoRegression eXogenous
BPTT	BackPropagation Through Time
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
CRAC	Computer Room Air Conditioner
DC	Data Center
DNN	Deep Neural Network
ECU	Electronic Control Unit
EKF	Extended Kalman Filter

EWC	Elastic Weight Consolidation
FCM	Fuzzy c-Means
FTC	Fault Tolerant Control
ICNN	Input Convex Neural Network
ICRNN	Input Convex Recurrent Neural Network
ICT	Information and Communication Technology
IoT	Internet of Things
ITE	IT Equipment
MISO	Multi-Input-Single-Output
MPC	Model Predictive Control
NLP	Natural Language Processing
OLS	Ordinary Least Square
PCA	Principal Component Analysis
PID	Proportional-Integral-Derivative
PLS	Partial Least Square
POD	Proper Orthogonal Decomposition
RBFNN	Radial Basis Function Neural Network
ReLU	Rectified Linear Unit

RMCU	Rack Mountable Cooling Unit
SGD	Stochastic Gradient Descent
SoC	State of Charge
SVM	Support Vector Machine
WSN	Wireless Sensors Network

Chapter 1

Introduction

1.1 Overview

With the rapid development of information and communication technology (ICT), increasing amount of IT equipments (ITEs) such as IT servers, storage devices and networking devices are required to support the services [1, 2]. The ITEs may be centralized within a facility called data center (DC), or decentralized into several DCs. The ITEs associated in DCs are easily managed and effectively operated [3, 4]. Except for the ITEs, supporting infrastructures including cooling system, power system and lighting system are also indispensable [5, 6]. Finally, the ITEs, supporting system, and necessary sensor network constitute the whole DC system [7].

As all the power consumption of ITEs in the DC would be dissipated in the form of heat, a cooling system is required to remove the heat from the indoor facility and release it outside [8, 9]. For the cooling system in DCs, several different cooling methods are developed which include air cooling, liquid cooling and two-phase cooling [10, 11, 12]. The difference between these three cooling methods is the heat transfer

medium. Although the thermal resistance of liquid cooling is much lower than that of air cooling, liquid cooling still cannot widely used for DCs due to the potential damage to electronics and complicated plumbing requirements [13, 14, 15]. Compared with air cooling and liquid cooling, two-phase cooling has higher energy loads [16]. However, the technical difficulty and high cost of two-phase cooling prevent its application. Therefore, air cooling almost dominates the market of cooling system in commercial DCs [17].

The location of cooling unit plays a significant role in cooling system as well. Generally, there are three different cooling architectures proposed in DCs that contain room-based, row-based and rack-based cooling architecture [18, 19]. Room-based cooling architecture means that the cooling system is mounted in a room and provides cold air to the whole room to cool the ITEs in DCs directly [20]. Row-based cooling architecture indicates that the cooling unit is installed in a row of connected racks and used for cooling the ITEs inside [21, 22, 23]. Rack-based cooling architecture is designed to cool the ITEs in each enclosed individual rack [24]. The configuration of three cooling architecture is shown in Figure 1.1.

The room-based cooling system is widely adopted in many DCs owing to the competitive price, convenient installation, and easy operation. However, with the growing construction of new DCs, the shortcoming of poor efficiency in room-based cooling system appears gradually [25]. Because of the hot air recirculation and cold air bypass in the facility, the room-based cooling system cannot cool the devices efficiently and thus causes the waste of much electricity [26, 27, 28]. Conversely, row-based and rack-based cooling system perform much better on energy efficiency [29]. The shorter airflow paths and distinct hot and cold chambers in DCs with enclosed

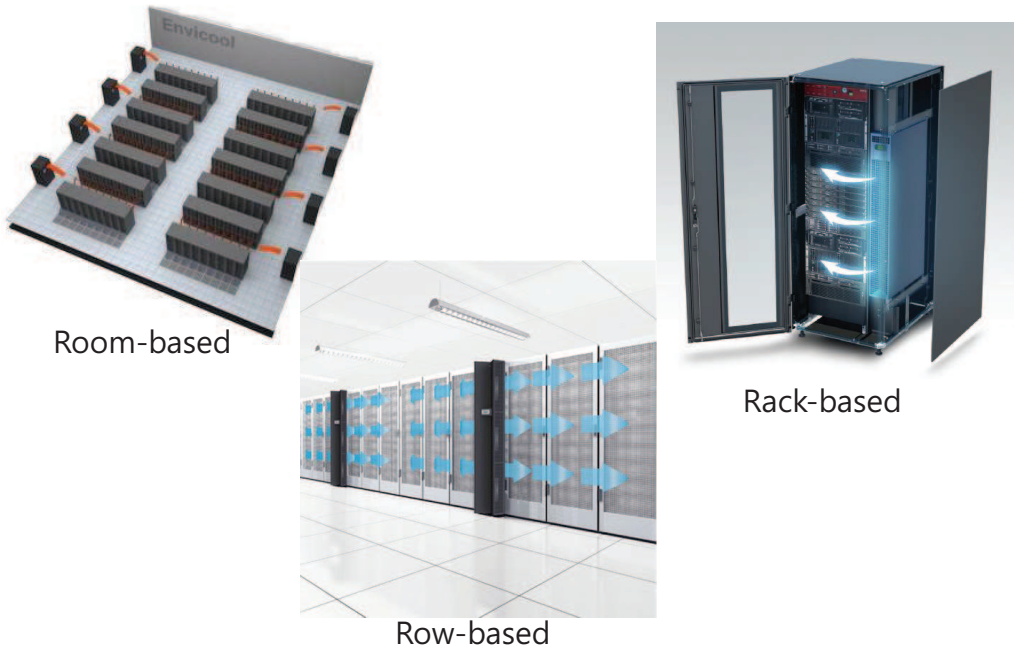

Figure 1.1: DCs equipped with room-based, row-based and rack-based cooling architecture

.

row-based and rack-based cooling architecture could significantly increase the cooling efficiency [30, 31]. Rack-based cooling is quite similar to row-based cooling, while rack-based cooling architecture with smaller container allows us to rapidly and precisely control the cooling load in response to varying workload of IT servers. Moreover, the economy, flexibility and scalability are also the advantages of rack-based cooling architecture [32]. Consequently, the research in this work mainly focuses on the DC equipped with rack-based cooling architecture.

Efficient temperature monitoring and control based on such cooling systems are also required to ensure the safe operation of ITEs [33, 34, 35]. In a DC, the security of ITEs should be of paramount importance. Exceeding the maximum operating temperature would result in ITE abnormal shut down, and further negatively affect

the health of ITEs [36]. Sometimes due to the improper airflow management and inadequate use of cooling capacity, hot spots may appear in the DC [37]. Hot spots are serious threats to the reliability and efficiency of ITEs. If the hot spots remain for a period of time, the ITEs would even be damaged irreversibly [38]. Therefore, a temperature monitoring system is needed to detect the hot spots and measure the temperature in DC constantly to ensure the ITEs operate within a suitable temperature range. Furthermore, temperature monitoring is also an effective approach to explore the overcooling, and thus avoid energy waste.

Only temperature monitoring system is not enough for the safety of DCs, while appropriate control strategies are also required to implement the temperature regulation [39]. At first, on-off control is used to regulate temperature, but it usually causes temperature oscillation in the facility [40]. The oscillation of temperature in DC would reduce the life span of ITEs [41]. Besides, the study in [36] reveals that temperature oscillation also greatly increases the risk of hard disk drive failure. Therefore, the efficient control strategies based on cooling systems are really necessary for guaranteeing the normal operation of DCs.

DCs in the US consumed nearly 70 billion kW· h of electricity in 2014, which accounts for two percent of the total power consumption in the US [42]. More seriously, the estimated power consumption of DCs would double in the next five years [43]. Among the total energy used in DCs, nearly forty percent is consumed by the cooling system [44]. The diagram of detailed energy consumption for each component in DC is presented in Figure 1.2. Initially, the cooling system without appropriate control in DCs commonly runs at full power, which would probably result in over cooling and energy waste [45]. On the other hand, if the cooling system runs at low

power, the ITEs would operate in high ambient temperature and usually consume more electricity. Thus, appropriate thermal management strategies with high cooling efficiency is also important for the energy management in the DCs.

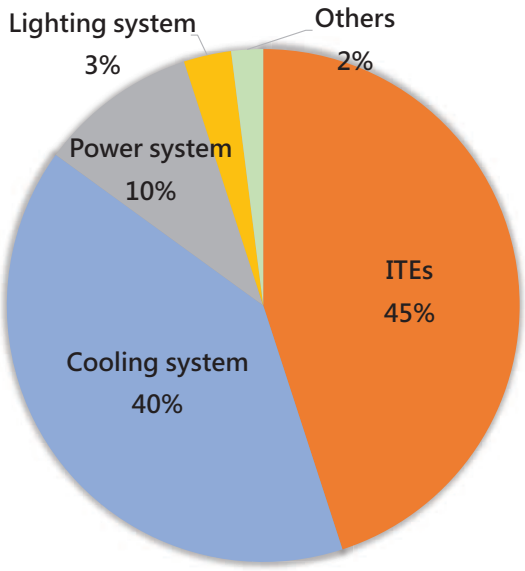

Figure 1.2: Fraction of energy consumption for each components in DC

.

1.2 Literature review

The areas of thermal management in DCs generally include thermal modeling, temperature monitoring, and temperature control. The thermal modeling is the fundamental of temperature monitoring and control, which is the representation of DC thermal dynamics. Temperature monitoring is designed to monitor or predict the temperature anomaly and warn the instructor. Finally, the temperature control is implemented to regulate the temperature in the DC and ensure the safe operation of

ITEs. For the thermal management in DCs, many investigations have been conducted.

1.2.1 Thermal Modeling and temperature prediction in DCs

Physical modeling

Physical model is one of the most common models for describing thermal dynamics in DCs. A simplified physical model was built for simulating the air flow and heat transfer in DC with room-based cooling architecture in [46]. In [47], an adaptive numerical temperature model was presented to capture the temperature behaviors in DCs. This model was built under the framework of state-space model and dedicated to controller design. Parolini et al. developed a thermal model in DC from a cyber-physical system perspective [48]. This paper constructed a thermal network based on thermodynamics and computational network respectively. Then these two models were coupled together and utilized for joint control in the DC. A transient thermal model was developed to capture the temporal thermal behaviors in [49]. This transient model was efficient to implement the temperature prediction and further task scheduling. Hosein et al. investigated real-time temperature modeling in a DC with rack mountable cooling unit (RMCU) [50]. In this paper, a novel transient zonal model was proposed based upon mass and energy conservation.

Physical modeling based on computational fluid dynamics (CFD) also attracted a lot of interest. A CFD model was studied to predict the thermal field for a small DC in [51]. The thermal boundary condition was particularly considered in this work, which reduced the prediction error greatly. CFD model was also used for transient temperature modeling with varying power dissipations and airflow rate in [52]. Chen

et al. studied a CFD model based on the realistic physical thermodynamics for temperature distribution forecasting [53]. The authors used the data from simulation and real experiments simultaneously, and significantly reduced the computational complexity of the proposed model. In [54], a CFD server level model was proposed to simulate the air flow rate and power fluctuations in a DC. This paper revealed that the thermal capacity of racks had small effect on the temperature change in DC. The authors in [55] proposed a CFD thermal model to detect the ITE failure and evaluate the thermal performance. This model described the detailed operational characteristics in DCs.

Data-driven modeling

Although the physical models are capable of representing the internal thermal dynamics of DCs, the model accuracy is not high enough due to the complicated heat transfer and air flow. To address this problem and reduce the modeling difficulty, data-driven models are widely utilized for thermal modeling of DCs. In [56, 57], the approach of proper orthogonal decomposition (POD) was employed to capture the relationships between temperature and major variables. Marina et al. employed grammatical evolution techniques for runtime temperature prediction in [58]. This method does not need specific knowledge about the thermal behaviors in DCs. The comparisons between CFD model and nonlinear ARX model indicated the effectiveness of the proposed model. Several data-driven techniques were adopted to predict the thermal anomaly in [59]. In this work, four methods including threshold method, moving averages-based method, exponentially weighed moving average-based method, and naive Bayesian classifier were compared for the anomaly prediction. Finally, the

simulations proved that the approach of naive Bayesian classifier had a better predictive accuracy. The authors employed a gradient boosting decision tree model to predict the rack temperature in [60]. The experiment results in this work illustrated that the proposed data-driven model has a high precision.

The statistical modeling frameworks show their superiority in temperature prediction, however, the poor extrapolative accuracy restricts the further application. Neural network is another effective method to implement temperature prediction in DCs. The authors in [61] utilized the artificial neural networks to predict the temperature in DC. Based on the artificial neural networks, a thermal-aware workload scheduling algorithm was proposed. In [62], the convolutional neural network (CNN) was used to predict the temperature distribution in DCs as well. Flavio et al. developed a temperature model based on neural networks to predict the real-time server inlet temperature in [63]. The neural network then was combined with a cooling system model to implement the strategy of temperature regulation in the DC. Zhang et al. adopted the artificial neural networks to predict the temperature and airflow simultaneously in [64]. Besides, the Latin Hypercube Sampling technique was utilized to determine design variables.

1.2.2 Temperature monitoring in DCs

Physical sensor is the most common tool for temperature monitoring in DCs. Since wired sensors are difficult to organize in such a small space, wireless sensors are widely used in DC [65]. The authors in [66] proposed an Internet of Things (IoT) system based on wireless sensors network (WSN) to monitor the temperature at different locations in DC. WSN was also employed to detect 3D temperature and humidity

distribution for DCs in [67]. The measured temperature and humidity were mainly utilized to improve the energy efficiency of DC. Polonelli et al. developed a low-cost and battery-supplied WSN for temperature monitoring in DCs [68]. The experimental results illustrated this type of WSN was able to provide fine-grained, flexible and long-term temperature monitoring.

Recently, mobile robots had been adopted for temperature monitoring in DCs gradually. Chris et al. developed an autonomous robot to implement temperature monitoring in DC [69]. The robot is able to generate the facility layout and thermal map in unseen DCs. After the mapping of DC, the robot could also detect the hot spots and other anomalies based on intelligent sampling. Choi et al. proposed a monitoring system in DCs based on multiple mobile robots in [70]. The superiority of this work was the cost efficiency and scalability of designed robots. Robots were designed for DC self-diagnosis in [71]. The robot was developed based on a series of real experiments and simulations and robust to the disturbances in real DC environments.

CFD model was employed to monitor temperature in DCs as well. The studies in [72, 73] investigated the construction of CFD models based on thermal mass and energy balance principles. The CFD models were not only use for temperature monitoring, but also for the prediction of energy consumption, airflow and pressure in DC. Finally, appropriate cooling strategies were determined according to the predictions of CFD models.

1.2.3 Temperature control methods in DCs

PID control

Proportional-integral-derivative (PID) control is a type of feedback control, which is widely used for process control in industry. The authors in [74] proposed a dynamic smart cooling controller to regulate the temperature in DC. Such controller was designed based on PID control and rule-based filter. The rule-based filter was used for identify the actuator faults and the PID control was used as the compensator of errors. A PID controller was designed in CFD environment to control the temperature in DC [75]. The proposed controller creatively considered the energy consumption of cooling system during the control process. In [76], a novel controller combining classic PID control and fuzzy adaptive control was developed to regulate the temperature in DC. The PID controller was regarded as the main controller to maintain the temperature, and the fuzzy adaptive control was employed to tune the parameters of PID control. By applying such method, the controller could deal with different working conditions in DC. Lee et al. put forward a self-tuning PID controller to regulate the temperature and optimize the energy consumption of fans simultaneously [77]. The PID controller was associated with a trained neural network, and thus could tune the parameters automatically.

Model predictive control

PID control was broadly applied in DC temperature control due to the simple development, while the low control stability and accuracy are still potential risks for the operation of ITEs. Therefore, to improve the control performance, model predictive control (MPC) is proposed to regulate temperature in DCs. MPC was proposed

to control the central processing unit temperature in servers for a modular DC in [78, 79, 80]. All the controllers were developed considering the energy consumption of cooling system. The authors in [81] designed a MPC to control the input air temperature and the number of active servers jointly. The numerical simulations revealed the MPC controller was efficient to reduce the power consumption in DC. Stochastic MPC was proposed to control the temperature in DC [82]. This thesis mainly focused on the control of cooling system and the servers's fans considering uncertainties in the forecasted IT workloads. Besides, the energy consumption was also designed as an important index in the stochastic MPC. SeyedMorteza et al. investigated the jointly controlling cooling units and workload assignment through MPC in [83]. The target of this paper was the optimization of energy consumption of DC while respecting temperature constraints. Additionally, a data-driven model was adopted to represent the thermal behaviors in DC. Parolini et al. also developed a MPC to manage the temperature and servers' workload in the view of cyber-physical system in [48].

Data-driven control

Except for MPC, some controllers based on data-driven techniques were studied for temperature regulation in DCs. Weiping et al. designed a combined neural network and genetic algorithm model for thermal management in DCs [84]. First, the thermal dynamics in DC were represented through a neural network. Then the genetic algorithm was used for optimizing the control inputs of cooling system. In [85], reinforcement learning algorithm was proposed to control the temperature and airflow in DC. The reinforcement learning without prior knowledge only needed a few hours for learning. Finally, the comparison between PID controller demonstrated the

effectiveness of the designed reinforcement learning.

1.3 Problem statement and research objectives

According to the above literature review, most of the works regrading to thermal modeling are conducted for the DCs with room-based cooling architecture, while the modeling of DCs with row-based and rack-based cooling has not been well studied. Nevertheless, the physical modeling is difficult to represent the thermal system in DCs accurately due to the complicated heat transfer and air flow. Therefore, the first objective of this research is developing efficient data-driven models including statistical models and neural networks to describe the thermal dynamics in DCs with rack-based cooling architecture.

Nearly all the thermal models in DCs are related to temperature prediction and control, while few researchers care about the real-time temperature estimation. Actually, the estimation of temperature distribution plays an essential role in reducing the cost of whole system and the burden of fault diagnosis in sensor networks. Therefore, the second objective of this research is to design effective observers based on the data-driven model to estimate the temperature distribution in DCs with rack-based cooling, and then extend this technique to row-based cooling architecture.

Based on the experiences in real industry, actuator faults of fans and valve usually occur in the daily operation. The actuator faults would significantly deteriorate the control performance, whereby threatening the safe operation of ITEs. Considering only a few basic research works are implemented in this field, the third objective of this research is to develop novel methods for actuator fault diagnosis and fault tolerant control in DCs.

As is shown in literature review, DNNs are widely adopted to conduct the thermal modeling and temperature prediction, owing to the powerful representation capability of nonlinear systems. However, most of the researches are just concentrated on the temperature prediction, and they fail to combine the DNNs with control issues. Therefore, the last objective of this research is to explore the construction of control-oriented DNNs and thus bridge the gap between control tractability and good representation ability of DNNs. In the future work, we would also validate the performance of control-oriented DNNs in the real DC experiments, and extend their applications in another industrial systems.

Bibliography

[1] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renew. Sustain Energy Rev.*, vol. 31, pp. 622–638, Mar. 2014.

[2] H. Zhang, S. Shao, H. Xu, H. Zou, and C. Tian, "Free cooling of data centers: A review," *Renew. Sustain Energy Rev.*, vol. 35, pp. 171–182, Jul. 2014.

[3] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (DCN): Infrastructure and operations," *IEEE communications surveys & tutorials*, vol. 19, no. 1, pp. 640–656, 2016.

[4] H. Lu, Z. Zhang, and L. Yang, "A review on airflow distribution and management in data center," *Energy and Buildings*, vol. 179, pp. 264–277, 2018.

[5] C. Nadjahi, H. Louahlia, and S. Lemasson, "A review of thermal management and innovative cooling strategies for data center," *Sustainable Computing: Informatics and Systems*, vol. 19, pp. 14–28, 2018.

[6] M. Wahlroos, M. Pärssinen, J. Manner, and S. Syri, "Utilizing data center waste heat in district heating–impacts on energy efficiency and prospects for

low-temperature district heating networks," *Energy*, vol. 140, pp. 1228–1238, 2017.

[7] T. Chen, X. Gao, and G. Chen, "The features, hardware, and architectures of data center networks: A survey," *Journal of Parallel and Distributed Computing*, vol. 96, pp. 45–74, 2016.

[8] Q. Zhou, J. Lou, and Y. Jiang, "Optimization of energy consumption of green data center in e-commerce," *Sustainable Computing: Informatics and Systems*, vol. 23, pp. 103–110, 2019.

[9] Y. Fu, W. Zuo, M. Wetter, J. W. VanGilder, and P. Yang, "Equation-based object-oriented modeling and simulation of data center cooling systems," *Energy and Buildings*, vol. 198, pp. 503–519, 2019.

[10] H. Moazamigoodarzi, S. Pal, D. Down, M. Esmalifalak, and I. K. Puri, "Performance of a rack mountable cooling unit in an it server enclosure," *Therm. Sci. Eng. Prog.*, vol. 17, p. 100395, Jun.

[11] T. Ding, Z. guang He, T. Hao, and Z. Li, "Application of separated heat pipe system in data center cooling," *Applied Thermal Engineering*, vol. 109, pp. 207–216, 2016.

[12] J. Cho, T. Lim, and B. S. Kim, "Viability of datacenter cooling systems for energy efficiency in temperate or subtropical regions: Case study," *Energy and buildings*, vol. 55, pp. 189–197, 2012.

[13] R. Gupta, S. Asgari, H. Moazamigoodarzi, S. Pal, and I. K. Puri, "Cooling architecture selection for air-cooled data centers by minimizing exergy destruction," *Energy*, p. 117625, Jun. 2020.

[14] J. Wang, Q. Zhang, S. Yoon, and Y. Yu, "Reliability and availability analysis of a hybrid cooling system with water-side economizer in data center," *Building and Environment*, vol. 148, pp. 405–416, 2019.

[15] Z. Li and S. G. Kandlikar, "Current status and future trends in data-center cooling technologies," *Heat Transfer Engineering*, vol. 36, no. 6, pp. 523–538, 2015.

[16] H. Moazamigoodarzi, "Distributed cooling for data centers: Benefits, performance evaluation and prediction tools," Ph.D. dissertation, 2019.

[17] S.-W. Ham, M.-H. Kim, B.-N. Choi, and J.-W. Jeong, "Simplified server model to simulate data center cooling energy consumption," *Energy and Buildings*, vol. 86, pp. 328–339, 2015.

[18] H. Moazamigoodarzi, R. Gupta, S. Pal, P. J. Tsai, S. Ghosh, and I. K. Puri, "Modeling temperature distribution and power consumption in it server enclosures with row-based cooling architectures," *Appl. Energy*, vol. 261, p. 114355, Mar.

[19] A. Capozzoli and G. Primiceri, "Cooling systems in data centers: state of art and emerging technologies," *Energy Procedia*, vol. 83, pp. 484–493, 2015.

[20] G. Callou, P. Maciel, D. Tutsch, and J. Araújo, "Models for dependability and sustainability analysis of data center cooling architectures," in *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, pp. 1–6. IEEE, 2012.

[21] C. Jin, X. Bai, J. Ni, J. Shen *et al.*, "Case study regarding the thermal environment and energy efficiency of raised-floor and row-based cooling," *Building and Environment*, vol. 182, p. 107110, 2020.

[22] K. Nemati, H. Alissa, and B. Sammakia, "Performance of temperature controlled perimeter and row-based cooling systems in open and containment environment," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 57502, American Society of Mechanical Engineers, 2015.

[23] J. Cho and J. Woo, "Development and experimental study of an independent row-based cooling system for improving thermal performance of a data center," *Applied Thermal Engineering*, vol. 169, p. 114857, 2020.

[24] A. Almoli, A. Thompson, N. Kapur, J. Summers, H. Thompson, and G. Hannah, "Computational fluid dynamic investigation of liquid rack cooling in data centres," *Applied energy*, vol. 89, no. 1, pp. 150–155, 2012.

[25] T. Ding, Z. guang He, T. Hao, and Z. Li, "Application of separated heat pipe system in data center cooling," *Appl. Therm. Eng.*, vol. 109, pp. 207–216, Oct. 2016.

[26] H.-J. Yin, Z. Yang, A.-Q. Chen, and N. Zhang, "Experimental research on a novel cold storage defrost method based on air bypass circulation and electric heater," *Energy*, vol. 37, no. 1, pp. 623–631, 2012.

[27] D. W. Demetriou and H. E. Khalifa, "Optimization of enclosed aisle data centers using bypass recirculation," *Journal of Electronic Packaging*, vol. 134, no. 2, 2012.

[28] S. Nada and M. Said, "Effect of CRAC units layout on thermal management of data center," *Applied thermal engineering*, vol. 118, pp. 339–344, 2017.

[29] S. Asgari, H. Moazamigoodarzi, P. J. Tsai, S. Pal, R. Zheng, G. Badawy, and I. K. Puri, "Hybrid surrogate model for online temperature and pressure predictions in data centers," *Future Generation Computer Systems*, vol. 114, pp. 531–547, Jan.

[30] S. Asgari, S. MirhoseiniNejad, H. Moazamigoodarzi, R. Gupta, R. Zheng, and I. K. Puri, "A gray-box model for real-time transient temperature predictions in data centers," *Appl. Therm. Eng.*, p. 116319, Nov.

[31] S. Nada and M. Said, "Comprehensive study on the effects of plenum depths on air flow and thermal managements in data centers," *International Journal of Thermal Sciences*, vol. 122, pp. 302–312, 2017.

[32] S. Shi, "Rack-based data center temperature regulation using data-driven model predictive control," Master's thesis, McMaster University, 2019.

[33] M. Truşcă, Ş. Albert, and M. Soran, "The benefits of data center temperature monitoring," in *Conference Grid, Cloud & High Performance Computing in Science (ROLCG)*, pp. 1–3. IEEE, 2015.

[34] B. Lajevardi, K. R. Haapala, and J. F. Junker, "Real-time monitoring and evaluation of energy efficiency and thermal management of data centers," *Journal of Manufacturing Systems*, vol. 37, pp. 511–516, 2015.

[35] J. Moore, J. Chase, K. Farkas, and P. Ranganathan, "Data center workload monitoring, analysis, and emulation," in *Eighth workshop on computer architecture evaluation using commercial workloads*, pp. 1–8. Citeseer, 2005.

[36] N. El-Sayed, I. A. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder, "Temperature management in data centers: why some (might) like it hot," in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, pp. 163–174, 2012.

[37] C. Guo, K. Xu, G. Shen, and M. Zukerman, "Temperature-aware virtual data center embedding to avoid hot spots in data centers," *IEEE Transactions on Green Communications and Networking*, 2020.

[38] P. Lin, "How to fix hot spots in the data center," *APC White Paper*, 2014.

[39] L. Parolini, B. Sinopoli, and B. H. Krogh, "Model predictive control of data centers in the smart grid scenario," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 10 505–10 510, Jan. 2011.

[40] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. Peng, "Green data center with IoT sensing and cloud-assisted smart temperature control system," *Computer Networks*, vol. 101, pp. 104–112, Jun. 2016.

[41] S. V. Patankar, "Airflow and cooling in a data center," *Journal of Heat transfer*, vol. 132, no. 7, 2010.

[42] H. Moazamigoodarzi, P. J. Tsai, S. Pal, S. Ghosh, and I. K. Puri, "Influence of cooling architecture on data center power consumption," *Energy*, vol. 183, pp. 525–535, Sep. 2019.

[43] S. MirhoseiniNejad, H. Moazamigoodarzi, G. Badawy, and D. G. Down, "Joint data center cooling and workload management: A thermal-aware approach," *Future Gener. Comput. Syst.*, vol. 104, pp. 174–186, Mar.

[44] T. Lu, X. Lü, M. Remes, and M. Viljanen, "Investigation of air management and energy performance in a data center in finland: Case study," *Energy and Buildings*, vol. 43, no. 12, pp. 3360–3372, Dec. 2011.

[45] H. Cheung, S. Wang, C. Zhuang, and J. Gu, "A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation," *Appl. energy*, vol. 222, pp. 329–342, Jul. 2018.

[46] V. Lopez and H. F. Hamann, "Heat transfer modeling in data centers," *Int. J. Heat Mass Transf.*, vol. 54, no. 25-26, pp. 5306–5318, 2011.

[47] V. A. Tsachouridis and T. Scherer, "Data centre adaptive numerical temperature models," *Trans. Inst. Meas. Control*, vol. 40, no. 6, pp. 1911–1926, 2018.

[48] L. Parolini, B. Sinopoli, B. H. Krogh, and Z. Wang, "A cyber–physical systems approach to data center modeling and control for energy efficiency," *Proc. IEEE*, vol. 100, no. 1, pp. 254–268, Jan. 2012.

[49] M. Jonas, R. R. Gilbert, J. Ferguson, G. Varsamopoulos, and S. K. Gupta, "A transient model for data center thermal prediction," in *2012 International Green Computing Conference (IGCC)*, pp. 1–10. IEEE, 2012.

[50] H. Moazamigoodarzi, S. Pal, S. Ghosh, and I. K. Puri, "Real-time temperature predictions in IT server enclosures," *Int. J. Heat Mass Transf.*, vol. 127, pp. 890–900, Dec. 2018.

[51] W. A. Abdelmaksoud, H. E. Khalifa, T. Q. Dang, R. R. Schmidt, and M. Iyengar, "Improved cfd modeling of a small data center test cell," in *12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 1–9. IEEE, 2010.

[52] M. Ibrahim, S. Bhopte, B. Sammakia, B. Murray, M. Iyengar, and R. Schmidt, "Effect of transient boundary conditions and detailed thermal modeling of data center rooms," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 2, no. 2, pp. 300–310, 2011.

[53] J. Chen, R. Tan, Y. Wang, G. Xing, X. Wang, X. Wang, B. Punch, and D. Colbry, "A high-fidelity temperature distribution forecasting system for data centers," in *2012 IEEE 33rd Real-Time Systems Symposium*, pp. 215–224. IEEE, 2012.

[54] S. Alkharabsheh, B. Sammakia, S. Shrivastava, and R. Schmidt, "Implementing rack thermal capacity in a room level cfd model of a data center," in *Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, pp. 188–192. IEEE, 2014.

[55] J. Cho, B. Park, and Y. Jeong, "Thermal performance evaluation of a data center cooling system under fault conditions," *Energies*, vol. 12, no. 15, p. 2996, 2019.

[56] E. Samadiani, Y. Joshi, H. Hamann, M. K. Iyengar, S. Kamalsy, and J. Lacey, "Reduced order thermal modeling of data centers via distributed sensor data," *ASME J. Heat Transfer*, vol. 134, no. 4, Feb. 2012.

[57] R. Ghosh and Y. Joshi, "Error estimation in POD-based dynamic reduced-order thermal modeling of data centers," *Int. J. Heat Mass Transf.*, vol. 57, no. 2, pp. 698–707, Feb. 2013.

[58] M. Zapater, J. L. Risco-Martín, P. Arroba, J. L. Ayala, J. M. Moya, and R. Hermida, "Runtime data center temperature prediction using grammatical evolution techniques," *Appl. Soft Comput.*, vol. 49, pp. 94–107, 2016.

[59] M. Marwah, R. Sharma, and C. Bash, "Thermal anomaly prediction in data centers," in *12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 1–7. IEEE, 2010.

[60] K. Sasakura, T. Aoki, M. Komatsu, and T. Watanabe, "Rack temperature prediction model using machine learning after stopping computer room air conditioner in server room," *Energies*, vol. 13, no. 17, p. 4300, 2020.

[61] L. Wang, G. von Laszewski, F. Huang, J. Dayal, T. Frulani, and G. Fox, "Task scheduling with ann-based temperature prediction in a data center: a simulation-based study," *Engineering with Computers*, vol. 27, no. 4, pp. 381–391, 2011.

[62] S. Tashiro, Y. Nakamura, K. Matsuda, and M. Matsuoka, "Application of convolutional neural network to prediction of temperature distribution in data centers," in *IEEE 9th International Conference on Cloud Computing (CLOUD)*, pp. 656–661. IEEE, 2016.

[63] F. De Lorenzi and C. Vömel, "Neural network-based prediction and control of air flow in a data center," *Journal of Thermal Science and Engineering Applications*, vol. 4, no. 2, 2012.

[64] S. Zhang, Y. Liu, W. Meng, Z. Luo, J. Bu, S. Yang, P. Liang, D. Pei, J. Xu, Y. Zhang *et al.*, "Prefix: Switch failure prediction in datacenter networks," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 1, pp. 1–29, 2018.

[65] M. Levy and J. O. Hallstrom, "A new approach to data center infrastructure monitoring and management (dcimm)," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–6. IEEE, 2017.

[66] S. Saha and A. Majumdar, "Data centre temperature monitoring with esp8266 based wireless sensor network and cloud based dashboard with real time alert system," in *2017 Devices for Integrated Circuit (DevIC)*, pp. 307–310. IEEE, 2017.

[67] M. G. Rodriguez, L. E. O. Uriarte, Y. Jia, K. Yoshii, R. Ross, and P. H. Beckman, "Wireless sensor network for data-center environmental monitoring," in *Fifth International Conference on Sensing Technology*, pp. 533–537. IEEE, 2011.

[68] T. Polonelli, D. Brunelli, A. Bartolini, and L. Benini, "A lorawan wireless sensor network for data center temperature monitoring," in *International Conference on Applications in Electronics Pervading Industry, Environment and Society*, pp. 169–177. Springer, 2018.

[69] C. Mansley, J. Connell, C. Isci, J. Lenchner, J. O. Kephart, S. McIntosh, and M. Schappert, "Robotic mapping and monitoring of data centers," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5905–5910. IEEE, 2011.

[70] W. Choi, K.-W. Park, and K. H. Park, "Scout: Data center monitoring system with multiple mobile robots," in *The 7th International Conference on Networked Computing and Advanced Information Management*, pp. 150–155. IEEE, 2011.

[71] J. Lenchner, C. Isci, J. O. Kephart, C. Mansley, J. Connell, and S. McIntosh, "Towards data center self-diagnosis using a mobile robot," in *Proceedings of the 8th ACM international conference on Autonomic computing*, pp. 81–90, 2011.

[72] N. Hassan, M. M. K. Khan, and M. Rasul, "Temperature monitoring and cfd analysis of data centre," *Procedia Engineering*, vol. 56, pp. 551–559, 2013.

[73] N. Ahuja, C. W. Rego, S. Ahuja, S. Zhou, and S. Shrivastava, "Real time monitoring and availability of server airflow for efficient data center cooling," in *29th*

IEEE Semiconductor Thermal Measurement and Management Symposium, pp. 243–247. IEEE, 2013.

[74] C. Bash, C. D. Patel, and R. K. Sharma, "Dynamic thermal management of air cooled data centers," in *Thermal and Thermomechanical Proceedings 10th Intersociety Conference on Phenomena in Electronics Systems*, pp. 8–pp. IEEE, 2006.

[75] B. Durand-Estebe, C. Le Bot, J. N. Mancos, and E. Arquis, "Data center optimization using PID regulation in CFD simulations," *Energy and Buildings*, vol. 66, pp. 154–164, 2013.

[76] J. Deng, L. Yang, X. Cheng, and W. Liu, "Self-tuning PID-type fuzzy adaptive control for CRAC in datacenters," in *International Conference on Computer and Computing Technologies in Agriculture*, pp. 215–225. Springer, 2013.

[77] C. Lee and R. Chen, "Optimal self-tuning PID controller based on low power consumption for a server fan cooling system," *Sensors*, vol. 15, no. 5, pp. 11 685–11 700, 2015.

[78] M. Ogawa, H. Endo, H. Fukuda, H. Kodama, T. Sugimoto, T. Horie, T. Maruyama, and M. Kondo, "Cooling control based on model predictive control using temperature information of it equipment for modular data center utilizing fresh-air," in *13th International Conference on Control, Automation and Systems*, pp. 1815–1820. IEEE, 2013.

[79] M. Ogawa, H. Endo, H. Fukuda, H. Kodama, T. Sugimoto, H. Soneda, and M. Kondo, "Cooling control restraining effects due to ict equipment utilization

of disturbance based on model predictive control for modular data center," in *IEEE Conference on Control Applications (CCA)*, pp. 183–190. IEEE, 2014.

[80] M. Ogawa, H. Fukuda, H. Kodama, H. Endo, T. Sugimoto, T. Kasajima, and M. Kondo, "Development of a cooling control system for data centers utilizing indirect fresh air based on model predictive control," in *7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 132–137. IEEE, 2015.

[81] M. Ogura, J. Wan, and S. Kasahara, "Model predictive control for energy-efficient operations of data centers with cold aisle containments," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 209–214, 2018.

[82] J. Olsson, D. Varagnolo, R. Lucchese, and J. Gustafsson, "Stochastic model predictive control for data centers," Ph.D. dissertation, Masters thesis, Luleå Uuniversity of Technology, 2016.

[83] S. Mirhoseininejad, G. Badawy, and D. G. Down, "A data-driven, multi-setpoint model predictive thermal control system for data centers," *Journal of Network and Systems Management*, vol. 29, no. 1, pp. 1–22, 2020.

[84] W. Yu, Z. Wang, Y. Xue, L. Guo, and L. Xu, "A combined neural and genetic algorithm model for data center temperature control." in *CIMA@ ICTAI*, pp. 58–69, 2018.

[85] N. Lazic, C. Boutilier, T. Lu, E. Wong, B. Roy, M. Ryu, and G. Imwalle, "Data center cooling using model-predictive control," in *Advances in Neural Information Processing Systems*, pp. 3814–3823, 2018.

Chapter 2

Temperature distribution estimation via data-driven model and adaptive Kalman filter in modular data centers

This chapter is reproduced from *"Temperature distribution estimation via data-driven model and adaptive Kalman filter in modular data centers"*, ***Kai Jiang***, *Shizhu Shi, Hosein Moazanigoodarzi, Chuan Hu, Souvik Pal and Fengjun Yan, Published in Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2020.* The author of this thesis is the first author and the main contributor of this publication.

2.1 Abstract

With the rapid development of information and communications technology (ICT), increasing number of data centers (DCs) are required to support the cloud computing, and critical web based services that run our daily lives. The conventional cloud DCs usually adopt computer room air conditioner (CRAC) or inRow units as the cooling sytem, while the rack mountable cooling unit (RMCU) is a more promising equipment due to the economy, exact controllability, flexibility and scalability. In order to ensure the efficiency of control system in RMCU and the security of servers in the DCs, the information of temperature distribution is very essential. Basically, the temperature distribution could be obtained through physical sensors easily. However, considering the cost of whole system and the burden of fault diagnosis in sensor networks, the number of temperature sensors should be kept down to a bare minimum. Therefore, it is necessary to develop an effective and real-time observer to estimate the temperature distribution in the system. Besides, due to the complex air flow and heat transfer in the container, it is quite difficult to construct a physics model. To this end, a novel observer embracing data-driven model and adaptive Kalman filter is proposed in this work. Auto regression exogenous (ARX) model is adopted as the framework of data-driven model, and the model is identified through a algorithm of partial least square (PLS). Moreover, to represent the nonlinear behaviors in the system, fuzzy c-means (FCM) is applied for data classification and getting multiple local linear models. Finally, adaptive Kalman filter is utilized to estimate the temperature distribution on the basis of proposed data-driven model. The estimation results based on experimental data indicate the performance of proposed approach is remarkable.

Key words: Data center, Temperature distribution, Data-driven model, Adaptive algorithm, Kalman filter.

2.2 Introduction

Exponential growth in the use of ICT globally has necessitated the establishment of DCs [1, 2]. A traditional DC is generally constructed in a huge plant with a number of supporting facilities [3]. Since all of the electrical energy consumed by the IT equipments in DC is eventually dissipated into heat, a cooling system is required to remove this heat from the DC and release it into the ambient air [4, 5]. Typically, conventional cooling systems such as CRAC are inefficient because of the air distribution inefficiencies (cold air bypass and hot air recirculation). To address this shortcoming, modular and distributed cooling systems are proposed and RMCU is the most efficient. As RMCU operates in the containment, it shortens the airflow paths and optimize the airflow distribution, and further improves the cooling efficiency. Additionally, the advantages of easy replacement, and maintenance in RMCU would also attract more attentions [6].

Meanwhile, to maintain stable temperature in the containment under varying IT equipment demand, superior control system for RMCU is required [7]. For such control system, accurate real time temperature distribution knowledge in the containment is necessary [8]. In addition, high temperature induced hot spots in the rack could also lead to IT equipments thermal shut down [9]. Thus, the temperature distribution is of critical importance to ensure the security in the system. Basically, several sensors are mounted in the rack to obtain the temperature distribution. However, considering the cost of whole system and the burden of fault diagnosis in sensor

networks, the number of temperature sensors should be kept down to a bare minimum. Consequently, the estimation of temperature distribution in DCs has aroused many interests among researchers [10, 11].

As the air flow and heat transfer in DCs is quite complex, it is difficult to establish physical model to estimate temperature distribution. Therefore, most of works about temperature distribution in DCs are based on data-driven methods. Neural network is employed to predict the temperature distribution in DC based on experimental and computational fluid dynamics (CFD) simulation data in [12]. The authors in [13] adopt a network model to predict the temperature distribution. They utilize machine learning technique to estimate the parameters in network model. Although the simulation results prove that the two approaches could predict the temperature distribution within small errors, the methods have weak robustness against the disturbances and noises. Besides, in [14] an adaptive numerical temperature model is developed to predict the temperature in DC. Robust recursive filtering algorithm is used to modified the temperature model based on real-time measurements form sensors, and the robustness of proposed model is improved. However, this technique fails to consider the complexity of sensor network in real applications.

There were also many works related to data-driven estimation recently. In [15], support vector machine (SVM) is employed to identify a Auto-Regressive Exogenous (ARX) model for fault diagnosis in the chillers and several cases of simulation have proved the outstanding performance of proposed approach. However, the model uncertainties and system disturbances aren't considered, which would degrade the accuracy of the proposed approach. Kang et al. study radial basis function neural network (RBFNN) model to deal with the problem of battery state of charge (SoC)

estimation. The comparisons between estimated SoC and modeled SoC demonstrate that the developed method could achieve a good accuracy and strong robustness. Nevertheless, the high computational complexity of neural network is a barrier to the practical situation. A high-order ARX model is also established in [17], where extended Kalman filter (EKF) is implemented to estimate solid oxide fuel cell stack temperature. The simulation results indicate that this method could get an accurate real-time estimation. However, the ARX-type model is just identified through the algorithm of extended least square, which cannot describe the nonlinearities of fuel cell system. What is more, an inaccurate model would also degrade the performance of Kalman filter.

Motivated by the problems, an original strategy based upon data-driven model and adaptive Kalman filter is developed for temperature estimation in the DC. For the modeling, ARX model is selected as the framework to describe the general dynamics in DC system. Then PLS is employed to calculate the parameters within the model. It is known that only a linear ARX model is not enough to represent the complex system, thus FCM is adopted to clustering the data set and multiple linear models are identified based on each cluster. At last, the multiple models are combined together through appropriate weights. By such technique of data-driven modeling, we can efficiently approximate the nonlinear behaviors and multivariate nature of system by linear models, and avoid complicated physical modeling. Furthermore, only the identified ARX model cannot estimate the temperature distribution precisely all the time, especially under noises and disturbances. Therefore, the Kalman filter is adopted to correct the estimations based on the outputs. Simultaneously, for the data-driven model, the process noise covariance is difficult to determine, thus, the

adaptive algorithm is adopted in Kalman filter to update the noise covariance step by step according to estimated errors and improve the accuracy of estimation substantially [18]. The outputs in this work are the temperature of two specific locations measured by sensors.

The main contributions of this work are summarized as follows. 1) A data-driven model based upon combined multiple local models, fuzzy c-means and partial least square is established to represent the nonlinearities in DC system. 2) Adaptive Kalman filter is employed to deal with the problems of varying noises in data-driven model and measurements respectively. 3) Real-time temperature distribution is estimated accurately with only a small amount of sensors, which is effective in guaranteeing the securities and reducing the cost in DCs. The rest of the article is organized as below. In Section 2, the detailed modular single-rack DC is described. Then the data-driven modeling and adaptive Kalman filter are represented in Section 3 and Section 4 respectively. Section 5 mainly depicts the simulation results and well-informed discussion. Finally, summary is shown in Section 6.

2.3 Fundamentals of modular single-rack data center

As is widely known, thermal management system is very important for the DCs. Generally, there are three locations to install the cooling system which could supply cool air to the servers, i.e., in the room, row, or rack [19]. Since the RMCU is mounted to or within the IT rack directly and delivers cold air to a single rack [20], the cold air flow path is shorter than that of room-mounted cooling and row-mounted cooling,

which could significantly reduce recirculation and avoid bypass [21]. Consequently, the full rated capacity of the rack-mounted cooling system can be utilized. Besides, the RMCU is more controllable and flexible than the others.

Since typical cooling unit can not provide necessary cooling capacity in a single rack [22, 23], for example a 5 kw telecommunication rack, an efficient RMCU is used in this work. The RMCU mainly consists of a chiller and cooling unit. The function of a chiller is to provide chilled water. Then the chilled water is pumped to the cooling unit located in the rack, which could cool the system by drawing warm air from the back to front through the coils filled with circulating chilled water [21]. At last, the heat generated by servers will be exhausted with the water exiting the rack and returning to the chiller. The whole system and cooling unit are represented in Figure 2.1 and Figure 2.2 respectively.

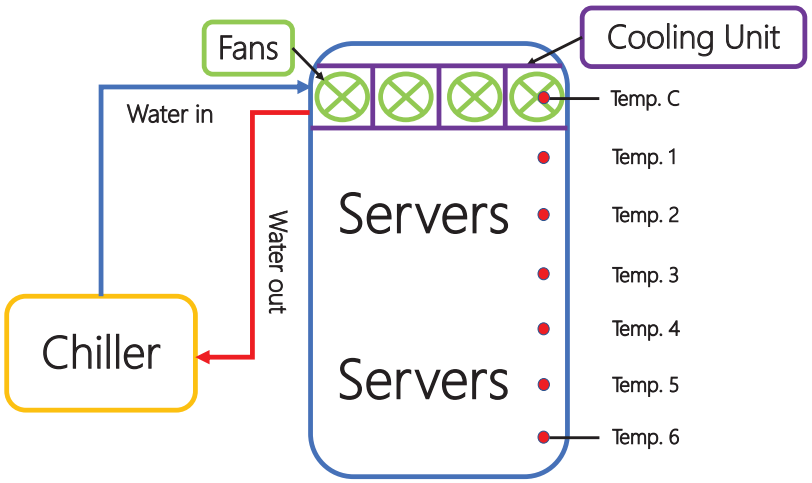

Figure 2.1: Structure of the modular single-rack data center.

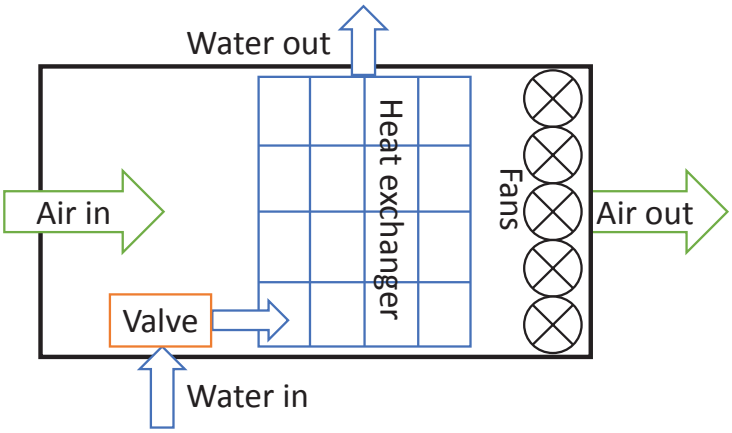

Figure 2.2: Rack-mounted water cooling system.

As is seen in Figure 2.1, the inputs of the system are water flow rate and air flow rate. The output are the temperature distributions in the front of the rack. The cooling unit is mounted in the top of rack, which includes fans, chilled water coil, power source, valve, input and output pipe.

2.4 Data-driven modeling

Since the severs cooling in an enclosed rack is always accompanied by complicated air flow and heat transfer, the modeling of thermal dynamics in data center is usually conducted via data-driven techniques. Regarding to data-driven modeling, ARX model is especially an appropriate framework for discrete dynamic systems [24]. In ARX model, the outputs at a specific sampling instant is represented by an equation concerning inputs, historical output information, historical input information and modeling bias. Nevertheless, ARX model can deal with multi-input-multi-output

(MIMO) system easily. The ARX model can be defined as below:

$$x(k) = \sum_{i=1}^{N_x} \alpha_i x(k-i) + \sum_{j=0}^{N_u} b_j u(k-j) + \theta + e(k), \tag{2.4.1}$$

where $x(k)$ denotes the output vectors at kth step and $u(k)$ means the input vectors. α_i, b_j and θ stand for coefficients in the model. $e(k)$ is the modeling bias. N_x and N_u represent the amount of time-lag in the outputs and inputs respectively.

As depicted in Eq. (2.4.1), the form of ARX model is just a linear model, which is not a very good description of nonlinear system. Therefore, it is necessary to identify numerous local models based upon the partitioned data [26]. For this work, FCM is adopted to clustering the data set, which is a well-known data clustering technique and evolved by c means algorithm initially. In FCM, all the data is divided into n clusters and each data point in the database belonging to each cluster with a certain degree. By minimizing the sum of weighted squared errors between every data point and cluster center, we can obtain the membership matrixes and cluster centers. Here, the cluster centers can be approximately regarded as the equilibrium points generally appear in nonlinear system [27]. Consequently, the nonlinear behaviors of thermal dynamics in data center could be described by the multiple linear models established through the data around each cluster efficiently. The core function of FCM is defined as below:

$$J_{FCM} = \sum_{i=1}^{N} \sum_{j=1}^{G} (\mu_{j,i})^q ||p_i - \sigma_j||. \tag{2.4.2}$$

Here, p_i= $[p_1, p_2, \cdots p_N]$ denote the data points in the vector space, N means the number of data points, σ_j represents the center of the jth cluster, G is the number

of clusters, μ_{ji} is named as the membership function which stands for the degree of p_i belonging to the jth cluster and weighted factor on each membership function is defined as q. Moreover, the objective function must satisfy some constraints, which is shown as follows:

$$\mu_{j,i} \in [0,1], \quad for \quad 1 \leq i \leq N, 1 \leq j \leq G, \tag{2.4.3}$$

$$\sum_{j=1}^{M} \mu_{j,i} = 1, \quad for \quad 1 \leq i \leq N, 1 \leq j \leq G, \tag{2.4.4}$$

$$0 < \sum_{i=1}^{N} \mu_{j,i} < N, \quad for \quad 1 \leq i \leq N, 1 \leq j \leq G. \tag{2.4.5}$$

The algorithm is carried out through an iterative minimization of the objective function through Lagrange multiplier [28]. First, the initializations of the number of clusters G, membership function μ_{ji} and weighted factor q are necessary to calculate the centers of cluster σ_j. Then according to the computed σ_j, membership μ_{ji} is updated and will be used for next step. The iteration will stop when

$$max_{ij}|\mu_{ji}(l+1) - \mu_{ji}(l)| < \xi, \tag{2.4.6}$$

where ξ is a termination criterion generally less than 1, and l is the iteration step. Besides, we can also set a certain l. When the iteration step reach the set number,

the iteration will stop. The results for iteration is shown as follows:

$$\mu_{ji} = \frac{1}{\sum_{k=1}^{G} (\frac{||x_i - c_j||}{||x_i - c_k||})^{2/(q-1)}}, \tag{2.4.7}$$

$$\sigma_j = \frac{\sum_{i=1}^{N} (\mu_{j,i})^q x_i}{\sum_{i=1}^{N} (\mu_{j,i})^q}. \tag{2.4.8}$$

In order to recognize the parameters in developed model, PLS is employed for this work. PLS regression is known as an efficient technique for figuring out the fundamental relationships between several variables[25]. Initially PLS is developed from ordinary least square (OLS) and mainly consists of the principal component analysis (PCA) and multiple regression [29]. PCA is a popular strategy for projecting no matter the dependent variables or independent variables from high dimensional space to a lower dimensional space, where the processed data could represent the underlying relations clearly and easier to deal with [30]. Then we can calculate the model parameters by utilizing OLS. For applying PLS conveniently, the ARX model is rewritten as follows:

$$Y = \beta X^{\mathrm{T}} + e(k), \tag{2.4.9}$$

where $Y = x(k)$ stands for a response matrix of all the output data, $X = [x(k-1)^{\mathrm{T}} \cdots x(k-N_y)^{\mathrm{T}} \quad u(k)^{\mathrm{T}} \cdots u(k-N_u)^{\mathrm{T}} \quad 1]^{\mathrm{T}}$ is defined as the regressor matrix, and $\beta = [a_1 \cdots a_{N_y} \quad b_1 \cdots b_{N_u} \quad c]$ represents the model coefficient matrix. The first step for PLS is to extract the principle components:

$$T = XP, \tag{2.4.10}$$

$$R = YQ, \tag{2.4.11}$$

in which T is defined as the principle component of X, P means the matrix of loadings of X and $||P||$=1. R denotes the principle components of Y, Q is the matrix of loadings of Y, and $||Q||$=1. As T is the principle component of X, the covariance of T itself should be maximum. Similarly, the covariance of R should be maximum as well. Besides, in order to explore a relationship between X and Y, we should also maximize the correlation of T and R. Then by applying Lagrange Multiplier and satisfying such conditions, the principle components T and R can be achieved. The equations are represented as below:

$$max < XP, YQ >, \quad s.t. \quad ||P|| = 1, ||Q|| = 1, \tag{2.4.12}$$

where $< XP, YQ >$ stands for the inner product of XP and YQ. As there exist modeling errors during the extraction of principle components, the Eq. (2.4.10) and Eq. (2.4.11) can be rewritten as:

$$X = TP^{\mathrm{T}} + E, \tag{2.4.13}$$

$$Y = RQ^{\mathrm{T}} + F, \tag{2.4.14}$$

where E is named as the matrix of residuals of X, and F stands for the matrix of residuals of Y.

As mentioned above, the correlation of T and R is maximized, thus the two vectors

could be linked by an inner relation. Further more, R can be substituted by T in this form:

$$Y = T\Phi^{\mathrm{T}} + \Omega, \tag{2.4.15}$$

where Φ is defined as the coefficient and Ω is the residual matrix. Since X, Y, T and R are all known now, we can obtain P, Q and Φ by minimizing the residual matrixes. Finally, we can obtain the coefficient matrix β_{PLS} and further get the identified ARX model:

$$\beta_{PLS} = P\Phi^{\mathrm{T}}. \tag{2.4.16}$$

$$Y = \beta_{PLS}X + e(k). \tag{2.4.17}$$

Finally, for describing the total nonlinearities in the system, all the identified models are associated together through several appropriate weights [26]. The schematic diagram of developed data-driven modeling is shown in Figure 2.3, and the combined ARX model is defined as below.

$$x(k) = \sum_{g=1}^{G} w_g(k)[\sum_{i=1}^{N_x} \alpha_{i,g}x(k-i) + \sum_{j=0}^{N_u} b_{j,g}u(k-j) + \theta_g], \tag{2.4.18}$$

where w_g is the weight of the gth model in the whole models, which is related to the membership function in FCM. c_g indicates the constant in the gth model. As all the identified linear models can potentially contribute to the prediction, the local models should be identified at the same time. The calculation can be described as below [26]:

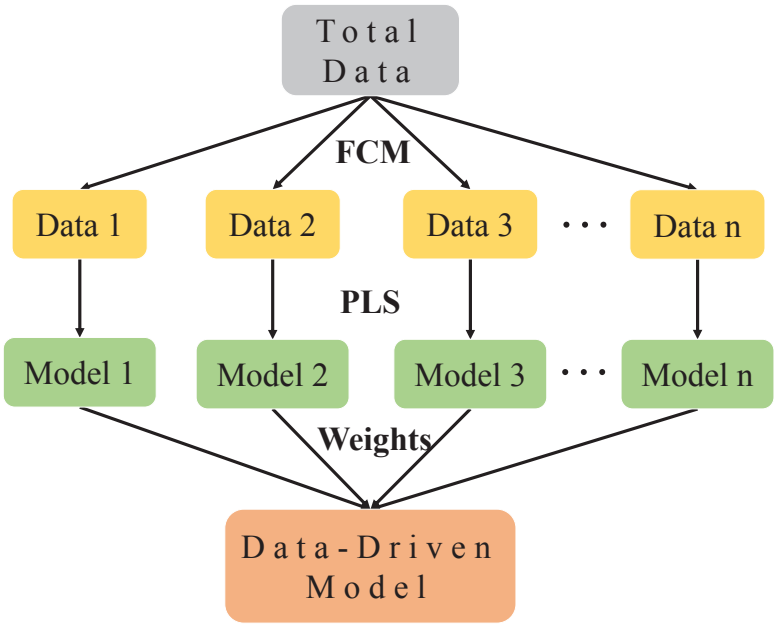

Figure 2.3: Schematic diagram of proposed data-driven modeling.

$$Y = [U_1 \otimes X \quad U_2 \otimes X \cdots U_g \otimes X] \times [\beta_1 \quad \beta_2 \cdots \beta_g]^{\mathrm{T}}. \tag{2.4.19}$$

Here, $\otimes$ depicts element by element multiplication, besides, U_g stands for the augment of membership matrix μ which is represented as follows:

$$U_g = \begin{bmatrix} \mu_{g,1} & \mu_{g,1} & \cdots & \mu_{g,1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{g,N} & \mu_{g,N} & \cdots & \mu_{g,N} \end{bmatrix}_{N \times G}. \tag{2.4.20}$$

In our system, the outputs $T(k)$ are the temperature distributions in the front of the rack. The inputs are the air flow rate F_{air} and water flow rate F_{wat} respectively. C is the constant term in the model, which is a vector too. What is more, the temperature of input water T_{in} and IT workload W are regarded as measurable

disturbances in the model. It should be pointed out that the temperature of input water should be constant, but it changes within 1 degree all the time due to the limitation of hardware (chiller). Therefore, the input water temperature is considered in the model identification. The system model is depicted as follows:

$$\begin{aligned} T(k) = &\sum_{i=1}^{N_x} \alpha_i T(k-i) + \sum_{j=0}^{N_{u,1}} b_{1,j} F_{air}(k-j) \\ &+ \sum_{l=0}^{N_{u,2}} b_{2,l} F_{wat}(k-l) + d_1 T_{in}(k) + d_2 W(k) + \Theta. \end{aligned} \tag{2.4.21}$$

According to the validated experiments for proposed data-driven model, N_x is selected as 1, $N_{u,1}$ is set as 0, and $N_{u,2}$ is chosen as 0 as well. Then the detailed model can be rewritten as below.

$$\begin{bmatrix} T_c(k) \\ T_1(k) \\ T_2(k) \\ T_3(k) \\ T_4(k) \\ T_5(k) \\ T_6(k) \end{bmatrix} = A \times \begin{bmatrix} T_c(k-1) \\ T_1(k-1) \\ T_2(k-1) \\ T_3(k-1) \\ T_4(k-1) \\ T_5(k-1) \\ T_6(k-1) \end{bmatrix} + B \times \begin{bmatrix} F_{air}(k) \\ F_{wat}(k) \end{bmatrix} + D \times \begin{bmatrix} T_{in}(k) \\ W(k) \end{bmatrix} + \begin{bmatrix} \hat{\theta} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}, \tag{2.4.22}$$

where A, B and D are all coefficients in the model. A is a 7×7 matrix. B and D are 7×2 matrixes.

2.5 Adaptive Kalman filter for temperature estimation

Although the combined ARX model could represent the global nonlinearities of the whole system, the noises, disturbances or some accidents in outside environment would deteriorate the accuracy of the identified model. Therefore, an observer is needed to correct the estimation online. Kalman filter is one of the most frequently-used and effective observers in engineering, which is found by Kalman and his collaborators decades ago [31]. Kalman filter is a method of optimal estimation by minimizing the estimated errors, and it could deal with the noises in system models and measurements. A general operation of Kalman filter for discrete system is shown below.

$$x(k) = \Gamma(k)x(k-1) + B(k)u(k) + w(k), \tag{2.5.1}$$

$$z(k) = \Pi(k)x(k) + v(k), \tag{2.5.2}$$

where Eq. (2.5.1) means a prediction model including the state-transition model $\Gamma(k)$, the input vector *u(k)*, state vector *x(k)* and Gaussian process noise vector *w(k)*. Similarly, Eq. (2.5.2) stands for an observation equation established through the observation model $\Pi(k)$, state vector, and unknown Gaussian measurement noise *v(k)*. Here both of *w(k)* and *v(k)* are assumed as zero-mean Gaussian noises, and their covariances are $\Lambda(k)$ and $\Psi(k)$ respectively. The procedure of Kalman filter mainly consists of two steps: one is prediction through system model and the other one is update via

estimated error. The equations for prediction are described as follows:

$$\hat{x}(k\,|k-1) = \Gamma(k)\hat{x}(k-1\,|k-1) + B(k)u(k), \tag{2.5.3}$$

$$P(k\,|k-1) = \Gamma(k)P(k-1\,|k-1)\Gamma(k)^T + \Lambda(k). \tag{2.5.4}$$

The prediction process is conducted through Eq. (2.5.3) and Eq. (2.5.4). Then the predicted states should be updated via a Kalman gain $\kappa(k)$ and the differences between calculations and sensor measurements *z(k)*, as the predictions of model are always effected by disturbances and noises. It is worthy note that the Kalman gain is obtained through the covariances of estimated errors *P(k)* which is derived from an initial value and updated iteratively. The detailed equations are represented as below.

$$\Upsilon(k) = \Pi(k)P(k\,|k-1)\Pi(k)^{\mathrm{T}} + \Psi(k), \tag{2.5.5}$$

$$\kappa(k) = P(k\,|k-1)\Pi(k)^{\mathrm{T}}\Upsilon(k)^{-1}, \tag{2.5.6}$$

$$\hat{x}(k\,|k) = \hat{x}(k\,|k-1) + K(k)[z(k) - \Pi(k)\hat{x}(k\,|k-1)], \tag{2.5.7}$$

$$P(k\,|k) = [I - \kappa(k)\Pi(k)]P(k\,|k-1). \tag{2.5.8}$$

According to the theory of Kalman filter, the identified ARX model is selected

as the prediction equation. The states in prediction equation are temperature of six locations in the rack, which is shown in Figure 2.1. The prediction equation is shown as followed.

$$\begin{aligned}\hat{x}(k\,|k-1) = \hat{T}(k\,|k-1) = \alpha\hat{T}(k-1\,|k-1) + b_1F_{air}(k) \\ + b_2F_{wat}(k) + d_1T_{in}(k) + d_2W(k) + \Theta.\end{aligned} \tag{2.5.9}$$

The measurement model can be represented as follows:

$$z(k) = \Pi(k)x(k). \tag{2.5.10}$$

Since the noises in this system cannot be invariable all the time, the noise covariance matrixes should be updated every time. Therefore, the adaptive algorithm is employed to calculate the the noise covariance matrixes online to promote the performance of Kalman filter. The adaptive algorithm is depicted as below:

$$\begin{aligned}\Lambda(k) = \Lambda(k-1) + \psi(k-1)[\kappa(k)\varepsilon(k)\varepsilon(k)^{\mathrm{T}}\kappa(k)^{\mathrm{T}} \\ + P(k) - P(k\,|k-1)],\end{aligned} \tag{2.5.11}$$

$$\Psi(k) = \Psi(k-1) + \psi(k)(\varepsilon(k)\varepsilon(k)^{\mathrm{T}} - P(k)), \tag{2.5.12}$$

$$\psi(k) = (1-b)/(1-b^{k+1}), \tag{2.5.13}$$

$$\varepsilon(k) = z(k) - \Pi(k)\hat{x}(k). \tag{2.5.14}$$

Here, $\psi(k)$ stands for a scaling parameter which is derived from a forgetting factor b. The function of forgetting factor is to increase the weight of 'new' data in the total data and it is commonly selected between 0.95 and 0.995. Additionally, ε_k refers to the estimation at kth step and the measured value from sensors.

To summarise, through the procedure of prediction, update and noise covariances calculation introduced above, the estimations of the temperature distribution based on the Kalman filter is completed. Furthermore, the whole algorithms used in this work are concluded in a flowchart and it is shown in Figure 2.4.

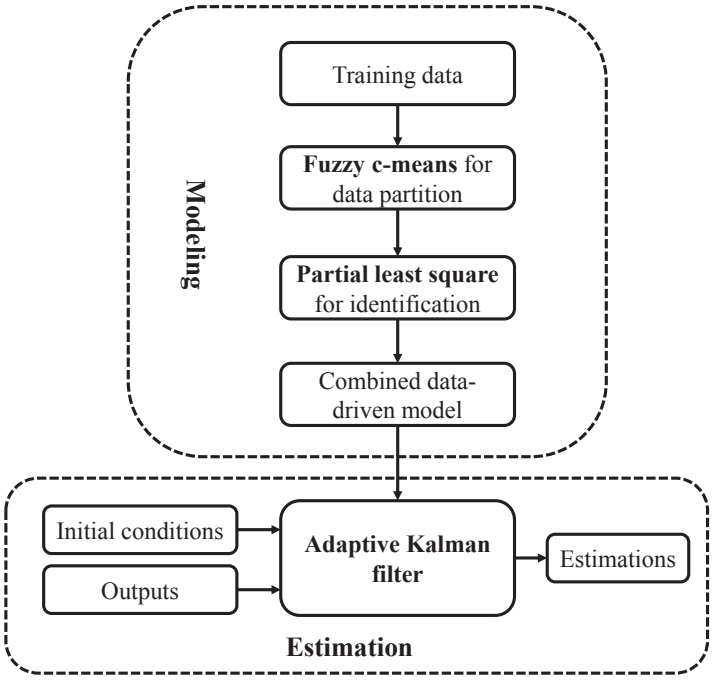

Figure 2.4: Flowchart of the algorithms in this work.

2.6 Experimental-data-based Simulation results

The data of overall working conditions used to develop the ARX model is based on the real experimental data from a single-rack DC. In each cycle of the experiment, the water flow rate and IT workload remain constant, and the air flow rate changes every one hour. This experiment consists of 10 different cycles and costs more than 80 hours. The setup is represented as Figure 2.5. The basic measurements cover the IT workload, temperature of input water, temperature in front of the rack, input water flow rate, and air flow rate are all measured during the whole test. The information is depicted in Figure 2.6.

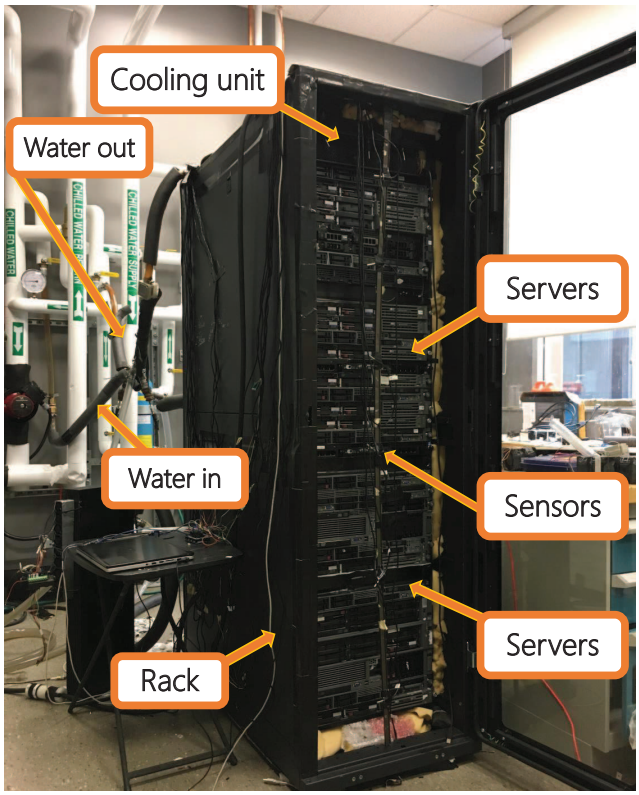

Figure 2.5: Single-rack data center with chilled water cooling system.

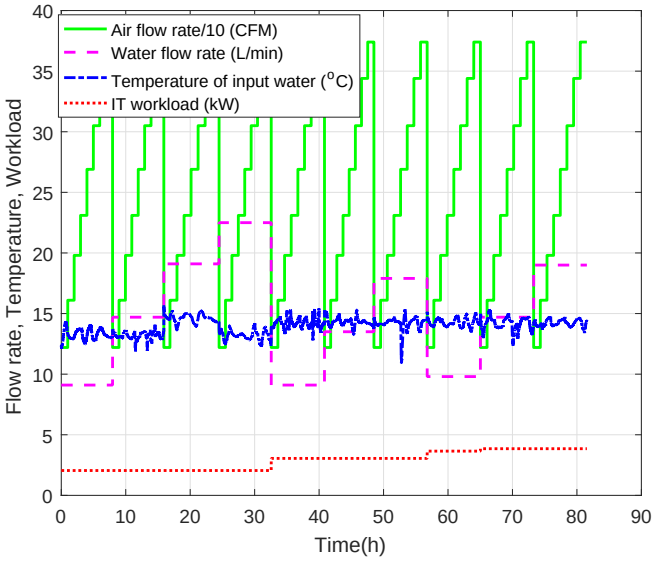

Figure 2.6: Measurements under all working conditions during the test.

2.6.1 Sensors placement

According to the simulation, if the temperature of only one location is used as the output, the estimate error would reach 2 degree, which is useless for practical application. If the temperature of three or more locations are selected as the output, the estimate accuracy improves significantly. However, too many sensors used in the rack are not economical. Thus, to make a trade-off between estimate accuracy and the number of temperature sensors, two sensors are employed as the outputs. First, the temperature of cooling unit ($Temp.C$) is the input cold air in front of the rack and it is essential for the temperature distribution, therefore, $Temp.C$ should be regarded as one of the outputs. Then the second output is determined by the simulation experiments. Specifically, the estimate errors based on all output combos ($Temp.C$ with $Temp.1$ to $Temp.6$) are compared to figure out the optimal sensors placement.

The average root mean square (ARMS) of errors are shown in Table 2.1.

Sensors combo	C1	C2	C3	C4	C5	C6
ARMS	2.25	1.59	1.47	1.56	1.45	1.43

Table 2.1: ARMS of estimate errors for temperature distribution under different sensor combos.

As we can see in the table, when $Temp.C$ and $Temp.6$ are used as the output to estimate the temperature distribution, the ARMS of error is 1.43 which is the smallest in all the errors. Therefore, we could conclude that the reasonable sensors placement is $Temp.C$ and $Temp.6$. Thus, $\Pi(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$.

2.6.2 Validation experiments

In order to validate the developed observer, another three experiments are carried out to obtain the data. In case I, the air flow and IT work load is kept as constant and the water flow rate changes per hour in each cycle and it takes about 3 hours. The measurements are represented in Figure 2.7. For the case II, the IT work load remains as a constant and the air flow and water flow change at the same time. The whole experiment costs 5 hours almost and all the measurements are shown in Figure 2.9. With regard to case III, all the variables including air flow, water flow and IT work load change simultaneously and the measurements are described in Figure 2.11.

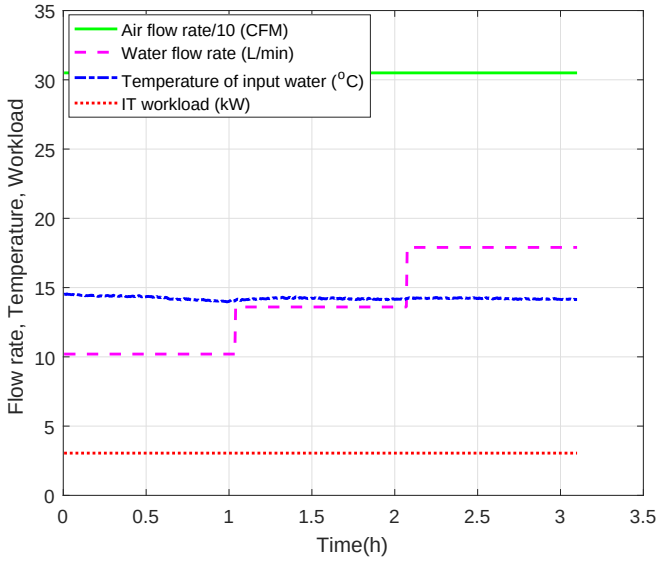

Figure 2.7: Measurements of validated experiment for case I.

The comparisons between experimental values, KF-based estimations and AKF-based estimations of temperature distributions in the first case are depicted in Figure 2.8. Since the temperature of cooling unit cannot reflect the state of servers and it is not our control target, the temperature is not represented in the figure. Besides, the temperature of cooling unit is quite similar to $Temp.1$, thus, the estimation of $Temp.1$ is very accurate. As described in the figure, although the differences between estimations of temperature and experimental value are a little big at the beginning, the estimations based on AKF could converge to the experimental value well toward the end of experimental time window. Despite the estimations of $Temp.3$ and $Temp.4$ are not as perfect as the others, they are also good enough for the industrial application. However, the estimated errors based on KF are much larger than that of AKF. Basically, the reason why the estimate errors of $Temp.3$ and $Temp.4$ are bigger than the other is the location of $Temp.3$ and $Temp.4$ is further to the selected sensors.

The root mean squares (RMS) of estimate errors are shown in Table 2.2.

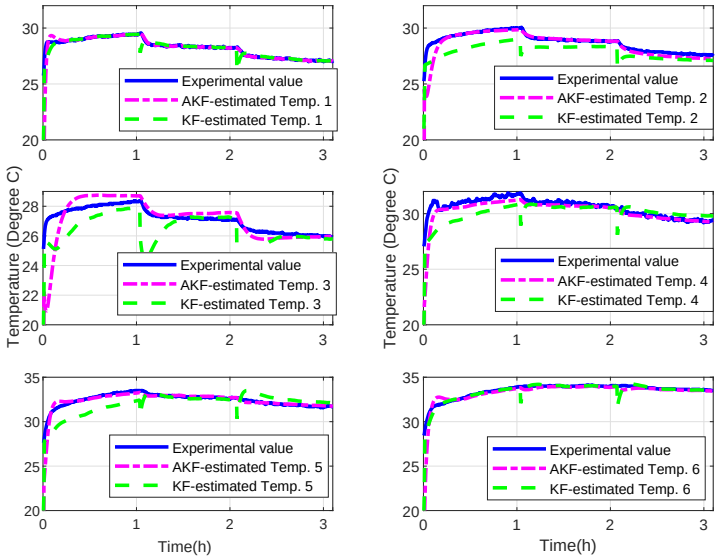

Figure 2.8: Comparison of temperature distribution between experimental values, KF-based estimations and AKF-based estimations in case I.

The information of the second validated experiment is displayed in Figure 2.9. The estimated temperature distributions compared with experimental values in the second case are described in Figure 2.10. As shown in the picture, all the estimations of temperature could follow the data in experiment well and converge to the actual value immediately. Although the estimate errors for $Temp.3$ and $Temp.4$ are a little larger than the others, they are still within 1 degree and it is acceptable for practical application. The RMS of errors are also represented in Table 2.2.

Figure 2.11 shows the whole measurements for case I. In this case, all the variables changes simultaneously, which is the most suitable to the actual situation in the data centers. The comparisons between temperature estimations and validated data are represented in Figure 2.12. In this figure, the performance of proposed observer is not

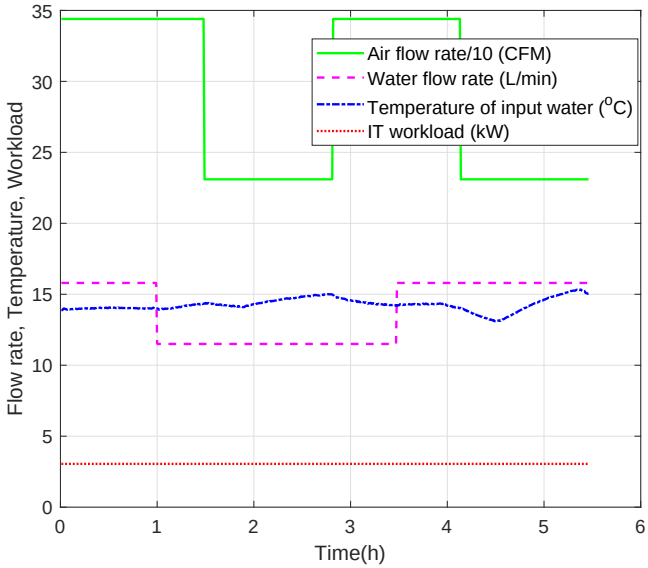

Figure 2.9: Measurements of validated experiment for case II.

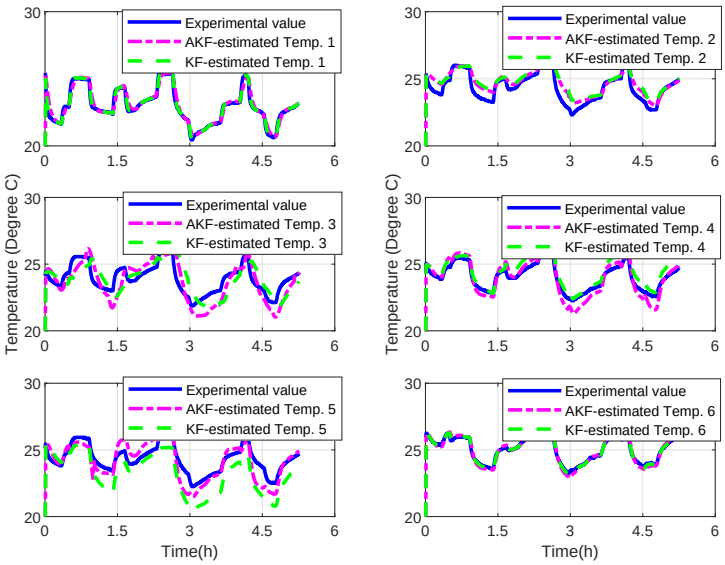

Figure 2.10: Comparison of temperature distribution between experimental values, KF-based estimations and AKF-based estimations in case II.

as good as that of case I and case II, as all the states in the system vary considerably. However, the estimations of $Temp.1$, $Temp.2$ and $Temp.6$ remain accurate as before. Additionally, the other estimations could track the transience of the experimental data, even though the estimate deviations are slightly larger than the deviations in case I and case II. The estimate errors are contained in Table 2.2 as well.

Table 2.2: RMS of estimate errors for temperature distribution in the rack.

Case	I	II	III
AKF-Temp.1	0.3434	0.3929	0.2643
KF-Temp. 1	1.4377	0.8538	0.7342
AKF-Temp.2	0.5737	0.5622	1.2526
KF-Temp. 2	4.4401	1.0630	1.3732
AKF-Temp.3	1.0026	1.2420	1.3577
KF-Temp. 3	1.1096	0.6559	1.2107
AKF-Temp.4	0.6185	0.4937	0.5481
KF-Temp. 4	1.1638	0.7156	1.1886
AKF-Temp.5	0.5740	1.6472	0.5481
KF-Temp. 5	1.5968	1.9740	0.6387
AKF-Temp.6	0.3929	0.2754	0.2194
KF-Temp. 6	1.0088	0.3036	0.1328

According to Table 2.2, almost all the RMS of estimate errors based on AKF are with 1.5 °C and less than that of KF, which indicates the proposed observer is effective enough to estimate the temperature distribution in the modular data center. Meanwhile, only two sensors are utilized to conduct the algorithm, thereby greatly optimizing the sensor network.

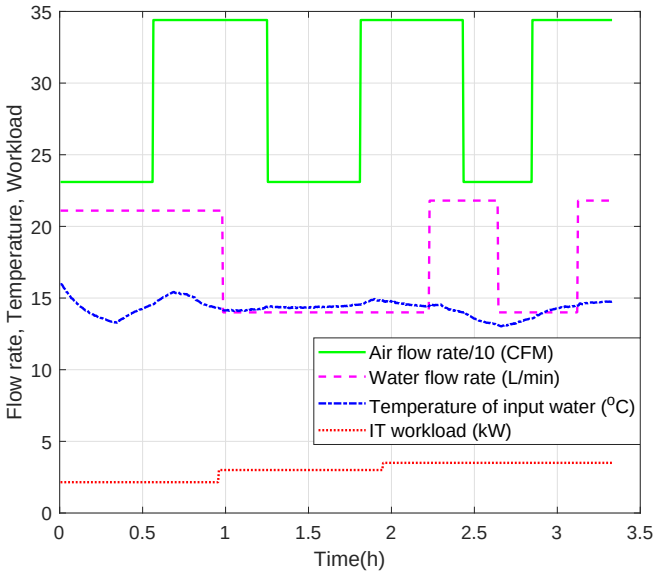

Figure 2.11: Measurements of validated experiment for case III.

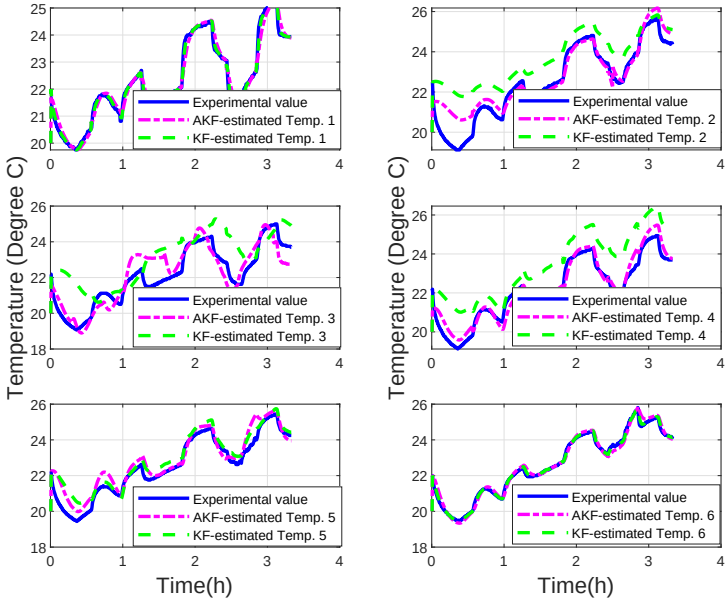

Figure 2.12: Comparison of temperature distribution between experimental values, KF-based estimations and AKF-based estimations in case III.

2.7 Conclusion

In this paper, an efficient estimator of temperature distribution is designed for a modular DC. The approach is developed upon the foundation of a data-driven model and adaptive Kalman filter. First Combined ARX model is established via FCM and PLS to describe the system. Then the adaptive Kalman filter is adopted to estimate the temperature distribution based on the measurements from only two temperature sensors in the rack. The comparisons between estimations and experimental values demonstrate the performance of proposed data-driven observer is excellent. In the future, the developed model and estimated temperature will be utilized in superior controllers. Besides, we will also focus on more data-driven modeling methods.

2.8 Acknowledgements

We acknowledge the funding support of National Sciences and Engineering Research Council (NSERC) of Canada and Cinnos Mission Critical Incorporated.

2.9 Appendix

Main Coefficients	Representation
N_x	Time-lag of the outputs
N_u	Time-lag of the inputs
e	Modeling bias
μ	Membership function
σ	Centers of cluster
T,R	Principle components
E,F,Ω	Residuals in PLS
w	Weights in data-driven model
F_{air}	Air flow rate
F_{wat}	Water flow rate
T_{in}	Temperature of input water
W	Workload of servers
$\Lambda(k)$	Covariance of procedure noise
$\Psi(k)$	Covariance of measurement noise
K	Kalman filter gain

Coefficients in data-driven model

$$A = \begin{bmatrix} 0.9998 & -0.0236 & -0.0280 & 0.0089 & 0.0088 & -0.0028 & -0.0048 \\ 0.0027 & 0.9728 & -0.0256 & 0.0088 & 0.0078 & -0.0026 & -0.0053 \\ 0.0038 & -0.0257 & 0.9705 & 0.0094 & 0.0085 & -0.0026 & -0.0052 \\ 0.0047 & -0.0258 & -0.0335 & 1.0080 & 0.0102 & -0.0037 & -0.0049 \\ 0.0093 & -0.0284 & -0.0358 & 0.0110 & 1.0081 & -0.0047 & -0.0051 \\ 0.0062 & -0.0262 & -0.0350 & 0.0099 & 0.0105 & 0.9946 & -0.0049 \\ 0.0004 & -0.0209 & -0.0193 & 0.0074 & 0.0057 & -0.0012 & 0.9935 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0093 & -0.0156 \\ -0.0080 & -0.0139 \\ -0.0072 & -0.0121 \\ -0.0047 & -0.0140 \\ -0.0060 & -0.0122 \\ -0.0069 & -0.0149 \\ -0.0170 & -0.0182 \end{bmatrix}$$

$$D = \begin{bmatrix} 0.0227 & 0.0077 \\ 0.0239 & 0.0077 \\ 0.0244 & 0.0075 \\ 0.0267 & 0.0091 \\ 0.0288 & 0.0094 \\ 0.0284 & 0.0090 \\ 0.0201 & 0.0056 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 0.7517 \\ 0.7284 \\ 0.7205 \\ 0.7892 \\ 0.7708 \\ 0.7667 \\ 0.6149 \end{bmatrix}$$

Bibliography

[1] L. Wang, SU. Khan, J. Dayal, "Thermal aware workload placement with task-temperature profiles in a data center," *Journal of Supercomputing,* vol. 61, no. 3, pp. 780-803, Sep. 2012.

[2] SW. Ham, MH. Kim, BN. Choi, JW. Jeong, "Energy saving potential of various air-side economizers in a modular data center," *Appl Energy,* vol. 138, pp. 258-275, Jan. 2015.

[3] H. Zhang, S. Shao, H. Xu, H. Zou, C. Tian, "Free cooling of data centers: A review," *Renew Sust Energy Rev,* vol. 35, pp. 171-182, Jul. 2014.

[4] J. Siriwardana, S. Jayasekara, SK. Halgamuge, "Potential of air-side economizers for data center cooling: a case study for key Australian cities," *Appl Energy,* vol. 104, pp. 207-219, Apr. 2013.

[5] SW. Ham, JS. Park, JW. Jeong, "Optimum supply air temperature ranges of various air-side economizers in a modular data center," *Appl Therm Eng*, vol. 77, pp. 163-179, Feb. 2015.

[6] H. Moazamigoodarzi, S. Pal, S. Ghosh, IK. Puri, "Real-time temperature predictions in IT server enclosures," *Int J Heat Mass Transf*, vol. 127, pp. 890-900, Dec. 2018.

[7] N. Rasmussen, "Guidelines for specification of data center power density," *APC White Paper,* 2005.

[8] Y. Ammar, S. Joyce, R. Norman, Y. Wang, AP. Roskilly, "Low grade thermal energy sources and uses from the process industry in the UK," *Appl Energy,* vol. 89, no. 1, pp. 3-20, Jan. 2012.

[9] J. Spitaels, "Dynamic power variations in data centers and network rooms," *APC White Paper,* 2011.

[10] K. Chen, C. Hu, X. Zhang, K. Zheng, Y. Chen, "Survey on routing in data centers: insights and future directions," *IEEE Network*, vol. 25, no. 4, pp. 6-10, July. 2011.

[11] C. Kelley, J. Cooley, "Deploying and using containerized/modular data center facilities," *The Green Grid White Paper*, vol. 42, 2011.

[12] J. Chen, R. Tan, Y. Wang, G. Xing, X. Wang, "High-fidelity temperature distribution forecasting system for data centers," in *IEEE 33rd Real-Time Systems Symposium*, pp. 215-224, Dec. 2012.

[13] S. Tashiro, Y. Tarutani, G. Hasegawa, et al, "A network model for prediction of temperature distribution in data centers," in *IEEE 4th International Conference on Cloud Networking*, Oct. 2015.

[14] A. Vassilios, S. Thomas, "Data centre adaptive numerical temperature models," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 6, pp. 1911-1926, Mar. 2017.

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

[16] LW. Kang, X. Zhao, J. Ma, "A new neural network model for the state-of-charge estimation in the battery degradation process," *Appl Energy*, vol. 121, pp. 20-27, May. 2014.

[17] A. Pohjoranta, M. Halinen, J. Pennanen, "Solid oxide fuel cell stack temperature estimation with data-based modeling-Designed experiments and parameter identification," *J Power Sources*, vol. 277, pp. 464-473, Mar. 2015.

[18] K. Jiang, E. Cao, L. Wei, "NOx sensor ammonia cross-sensitivity estimation with adaptive unscented Kalman filter for Diesel-engine selective catalytic reduction systems," *Fuel*, vol. 165, pp.185-192, Feb. 2016.

[19] K. Ebrahimi, GF. Jones, AS. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renew Sust Energy Rev*, vol. 31, pp.622-638, Mar. 2014.

[20] Y. Fulpagare, A. Bhargav, "Advances in data center thermal management," *Renew Sust Energy Rev*, vol. 43, pp.981-996, Mar. 2015.

[21] T. Evans, "The Different Technologies for Cooling Data Centers," *APC White Paper,* 2011.

[22] AH. Khalaj, SK. Halgamuge, "A Review on efficient thermal management of air- and liquid-cooled data centers: From chip to the cooling system," *Appl Energy*, vol. 207, pp. 1165-1188, Nov. 2017.

[23] M. Bramfitt, H. Coles, "Modular/container data centers procurement guide: optimizing for energy efficiency and quick deployment," *Lawrence Berkeley National Laboratory,* Feb. 2011.

[24] F. Ferracuti, A. Fonti, L. Ciabattoni, S. Pizzuti, A. Arteconi, "Data-driven models for short-term thermal behaviour prediction in real buildings," *Appl Energy*, vol. 204, pp. 1375-1387, Oct. 2017.

[25] S. Aumi, P. Mhaskar, "An adaptive data-based modeling approach for predictive control of batch systems," *Chem Eng Sci*, vol. 91, pp. 11-21, Mar. 2013.

[26] S. Aumi, B. Corbett, P. Mhaskar, "Data-based modeling and control of Nylon-6, 6 batch polymerization," *IEEE Trans Cont Syst*, vol. 21, no. 1, pp. 94-106, Jan. 2013.

[27] S. Aumi, P. Mhaskar, "Integrating data-based modeling and nonlinear control tools for batch process control," *AIChE J*, vol. 58, no. 7, pp. 1374-1386, Jul. 2012.

[28] NR. Pal, K. Pal, JM. Keller, "A Possibilistic Fuzzy C-Means Clustering Algorithm," *IEEE Trans Fuzzy Syst*, vol. 13, no. 4, pp. 517-530, Aug. 2005.

[29] YP. Zhou, JH. Jiang, WQ. Lin, L. Xu, HL. Wu, GL. Shen, "Artificial neural network-based transformation for nonlinear partial least-square regression with application to QSAR studies," *Talanta*, vol. 71, no. 2, pp. 848-853, Feb. 2007.

[30] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction, *IEEE Trans Auto Cont*, vol. 26, no. 1, pp. 17-32, Feb. 1981.

[31] K. Jiang, P. Geng, F. Meng, H. Zhang, "An extended Kalman filter for input estimations in diesel-engine selective catalytic reduction applications," *Neurocomputing*, vol. 171, pp. 569-575, Jan. 2016.

Chapter 3

Data-driven fault tolerant predictive control for temperature regulation in data center with rack-based cooling architecture

This chapter is reproduced from *"Data-driven fault tolerant predictive control for temperature regulation in data center with rack-based cooling architecture"*, ***Kai Jiang***, *Masoud Kheradmandi, Chuan Hu, Souvik Pal and Fengjun Yan, submitted to Mechatronics, 2020.* The author of this thesis is the first author and the main contributor of this publication.

3.1 Abstract

Recently increasing amount of data centers (DCs) have been constructed for the development of information and communications technology (ICT). Meanwhile, more energy consumption is required to support the operation of DCs, where nearly thirty percent of the consumption is used for cooling system. Therefore, this paper studies an efficient cooling architecture named rack mountable cooling unit (RMCU) for DCs, and provides a novel data-driven fault tolerant predictive controller to regulate the temperature accordingly. In order to circumvent the complicated physics modeling in DC, a data-driven modeling method is developed. In this method, multiple locally linear models in form of auto-regressive exogenous (ARX) model are selected to describe the strongly nonlinear system. Besides, the multiple sub-models are identified through partial least square (PLS) and corresponding training data is partitioned by fuzzy c-means (FCM). Then, on the basis of developed data-driven model, a predictive controller considering actuator faults is designed to manage the temperature in DC. Finally, real experimental results are presented and demonstrate the superior performance of our proposed controller.

Key words: Data center, Temperature control, Data-driven model, Fault tolerant control.

3.2 Introduction

With the growing demand of novel ICT, more DC facilities are built to maintain the operation and more energy consumption is required accordingly [1]. It was reported

that the power consumption of DCs was nearly 1.3 % of the total global electricity in 2010, and 30% of the consumption in DCs was supplied to cooling system [2]. Therefore, appropriate cooling architectures and control strategies are needed to improve the energy efficiency in cooling system. Generally, there are three cooling structures in DC including room-based, row-based and rack-based cooling. Here room-based cooling means the cold air is supplied to the computer room directly and cool down the whole servers; row-based cooling denotes that several cooling units are mounted between the racks and each of them delivers cold air to chill a row of servers; rack-based cooling indicates the cooling unit is installed within a single rack [3]. The schematic diagram of three different cooling architectures is shown in Figure 3.1. Nowadays room-based cooling architecture is widely used in DCs due to the simplicity of manufacturing and handling, while the problems of cold air bypass and hot air recirculation in such DCs deteriorate the energy efficiency significantly [4]. Therefore, row-based and rack-based cooling architectures are developed and become prevalent. The shorter and more predictable airflow paths in these cooling methods are able to address the above problems and reduce the power consumption [5].

Except for the development of cooling architecture, control design of regulating temperature in DCs is also a critical and challenging issue [6, 7, 8]. Firstly, the DC system is a nonlinear multi-input-single-output (MISO) system, and thus advanced controller is necessary to ensure the temperature stability. Moreover, the complicated airflow and heat transfer in DC further increase the difficulty of modeling and control. Therefore, many researchers are attempting to explore efficient control strategies to overcome the drawbacks, and some relevant works about cooling system control have been carried out so far.

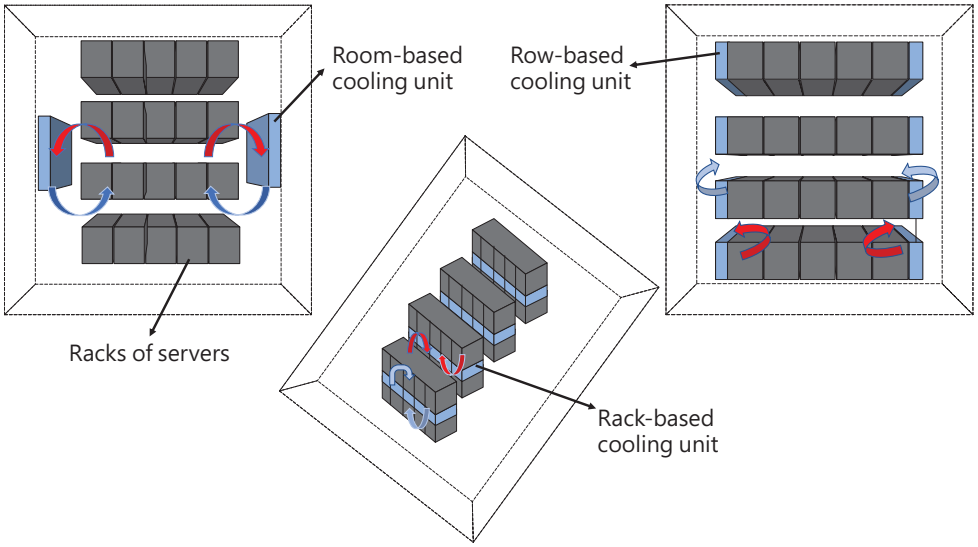

Figure 3.1: Schematic diagram of DC with room-based, row-based and rack-based cooling architecture

.

The initial control strategy used in cooling system is on-off control, which means the cooling devices would be turned off or on when the temperature below or above the set point. This on-off control is the simplest control in cooling system, while the frequently on-off would lead to temperature instability and reduce the lifetime of cooling equipments [9]. Proportional-integral-derivative (PID) control is another effective technique for cooling system control in DCs. For instance, Baptiste et al. designed a PID controller for DC optimization in [10], where the controller was mainly utilized to manage the air speed. Self-tuning PID control was proposed to control CRAC in DCs as well [11]. In this work, the PID control was adopted to regulate temperature automatically and a fuzzy logic method was used to tune the parameters in PID controller. Although PID control had achieved good effectiveness in cooling system, the problem of large temperature oscillation was still a threat for server security and

probably resulted in energy inefficiency. Lately, model predictive control (MPC) in DC cooling system has attracted a great deal of interests. Decentralized MPC was employed to control the blower speed and supplied water temperature in multiple CRAC units [12]. This method was able to meet the constraints of temperature in each zone and minimize the power consumption of CRAC at the same time. A novel controller based on MPC was proposed for cooling system in [13], in which indirect fresh air was utilized and the power consumption was reduced by 30%. Besides, the authors in [14] investigated DC control in cyber-physical perspective through MPC. This work considered computational network, thermal network and the effect of electricity market, and MPC was used to implement the global optimization in DC. At last the simulation indicated this method could achieve great cost saving. As discussed above, MPC is extensively applied in DCs and satisfactory result is achieved. However, most of the existing works concerning MPC in DCs are limited by physics modeling, which is quite complicated and time-consuming. Moreover, inaccurate physics models owing to many modeling assumptions usually lead to the decrease of control performance.

To tackle the problems, data-driven predictive control is put forward for DC with rack-based cooling architecture. In order to simulate the nonlinear characteristics in DC system, the data-driven model is established on multiple linear ARX models which is identified via PLS. Besides, the algorithm of FCM is utilized to partition the training data for multiple modeling. Finally, appropriate weights derived through membership matrix in FCM are used to combine the multiple sub-models together. According to such data-driven model, predictive controller is developed to implement the temperature regulation in DC. Additionally, fault tolerant strategy is creatively

considered in the data-driven control to deal with three typical actuator faults (aging of fans, failure of fans and valve deactivation) in rack-based cooling architecture.

The main contributions in this work are illustrated as follows. 1) A novel data-driven model based on the algorithm of FCM and PLS is established to represent the nonlinear dynamics in DC system. 2) Three types of actuator faults are considered in this work, and corresponding fault factors are identified during the control process to compensate the faults. 3) Different cost functions are switched for the data-driven MPC framework according to the three actuator faults and thus improving the control performance.

The rest of this article is organized as following. Section II presents the fundamental of DC and the rack-based cooling system. Then the methodology of data-driven modeling is introduced in Section III. The detailed development of fault tolerant predictive control and experimental results in real system are provided in Section IV and Section V separately. Finally, conclusion is depicted in VI.

3.3 Description of DC with rack-based cooling architecture

Cooling system plays a significant role in DC to regulate temperature and further protect servers. Currently, there are three types of cooling methods in DCs, which contains air cooling, liquid cooling and phase-change-material cooling [15]. Considering the operational feasibility, cooling security and equipment cost, the method of air cooling is widely adopted in DC cooling system. In light of cooling architecture, room-based, row-based and rack-based cooling units are regarded as three

typical architectures. As mentioned above, row-based and rack-based cooling system are more energy-efficient over room-based cooling system. Besides, in row-based and rack-based cooling systems, rack-based cooling system is more suitable to DCs with stand-alone high-density racks due to the flexibility and scalability. Thus, the detailed investigation relating to rack-based cooling architecture is conducted in this work.

The main concept of rack-based cooling system in DCs is to extract the heat generated by servers out of rack through the heat exchange between chilled water and hot air. The airflow from hot aisle to cold aisle is driven by several fans and goes through a heat exchanger filled with circulating chilled water, then the heat of hot air would be taken away via the chilled water. The circulating water is supplied and cooled through a chiller outside. The whole system is depicted in Figure 3.2. The RMCU is also represented in Figure 3.3, which comprises fans, heat exchanger, valve, power source and some pipes. Moreover, the location of RMCU in the rack could be adjusted based on the demand of customers.

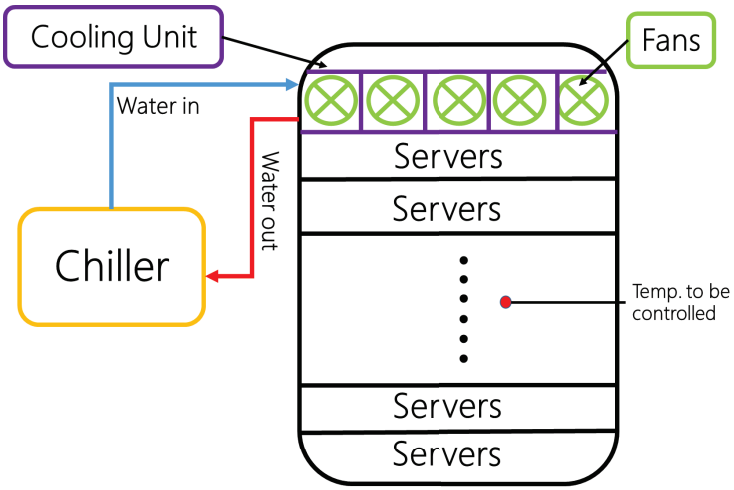

Figure 3.2: Detailed configuration of DC with rack-based cooling architecture.

According to Figure 3.2 and Figure 3.3, we can figure out the thermal control system in DC clearly. The measured plant output or the control objective is the temperature in rack. The inputs to the plant or the outputs of the controller are the air flow rate controlled by fans and the water flow rate controlled by valve. In addition, the temperature of inlet water is considered as measured disturbance, which is determined by the chiller system in the building.

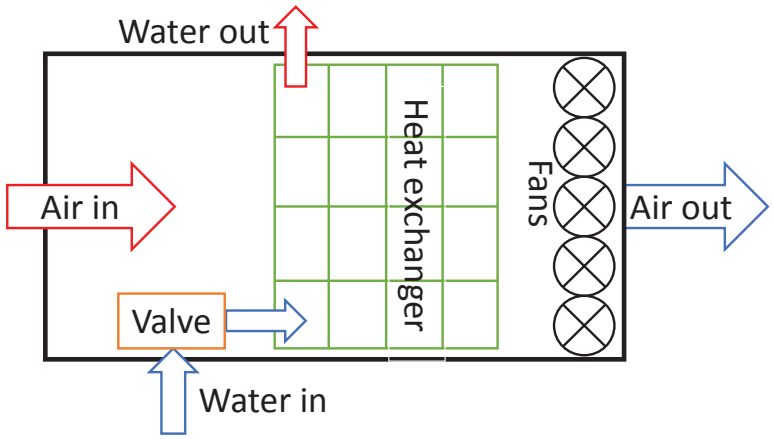

Figure 3.3: Internal structure of rack mountable cooling unit.

3.4 Data-driven modeling

Although the individual rack is just a small part of DC, the thermal dynamics inside including heat conduction, convection, and radiation are quite complicated and thus difficult to model through first principle. Moreover, the different types of servers in the rack have different thermal characteristics, which would further increase the modeling difficulty. Therefore, a data-driven model is developed in this program to tackle the modeling problem and used for further predictive control.

The framework of data-driven model is based on ARX model, which is an efficient linear discrete model to represent the relationship between inputs and outputs [16]. The typical ARX model is formulated as following. As seen in Eq. (3.4.1), the output at specific step is a function of historical outputs, current inputs, historical inputs, constant factor and modeling bias. $x(k)$ in the equation denotes output vector at kth step, $u(k)$ stands for input vector, and $b(k)$ means the modeling bias. The parameters α_i and β_i are system coefficients. σ represents the constant factor. N_x and N_u express the total number of time lapse in historical outputs and inputs.

$$x(k) = \sum_{i=1}^{N_x} \alpha_i x(k-i) + \sum_{j=0}^{N_u} \beta_j u(k-j) + \sigma + b(k), \tag{3.4.1}$$

The algorithm of PLS is utilized for parameter identification in ARX model in this work, which is a popular multivariate statistical method of data-driven modeling. The main procedure of PLS comprises two steps: one is data preprocessing and the other one is model identification [17, 18]. The data preprocessing is conducted through the approach of principle component analysis (PCA) that has the ability of projecting the latent variables in high-dimension to low-dimension. The parameter identification is carried out by a typical method of regression called ordinary least square (OLS). Owing to the superiority of PCA, the method of PLS is quite suitable to handle the modeling issue with high correlation. The main equations in PLS are expressed as following.

$$X = T_X P^{\mathrm{T}} + E_X, \tag{3.4.2}$$

$$Y = T_Y Q^{\mathrm{T}} + E_Y, \tag{3.4.3}$$

where X and Y represent inputs and outputs, T_X and T_Y stand for principle component of inputs and outputs respectively, E_X and E_Y denote the model residuals of X and Y, P means the matrix of loadings of X and Q is the matrix of loadings of Y. Then, to find the relationship between inputs and outputs, a function linked two principle components is required:

$$T_Y = ST_X, \tag{3.4.4}$$

where S is the parameter of two principle components. At last, by applying Lagrange multiplier, all the parameters are derived and the final relationship of inputs and outputs is demonstrated as below:

$$Y = \theta X + b(k), \tag{3.4.5}$$

in which θ is defined as the model parameters. Besides, if the objective function is organized as the form of ARX model, model parameters can be depicted as: $\theta = [\alpha_1 \cdots \alpha_{N_y} \quad \beta_1 \cdots \beta_{N_u} \quad \sigma]$ and $X = [x(k-1)^T \cdots x(k-N_y)^T \quad u(k)^T \cdots u(k-N_u)^T \quad 1]^{\mathrm{T}}$.

As the thermal dynamic in DC is a quite complex process, only a linear ARX model is insufficient to represent the system. Additionally, the complicated system is later found to be locally linear in short time series. Thus, multiple local linear models are put forward to approximatively simulate the nonlinearities in DC system. In order to obtain multiple models, the dataset for training is divided into some clusters and then the model parameters are calculated on the basis of each cluster. In this article, the approach of FCM is used for clustering, which is a typical method of clustering

and widely adopted for data processing, image segmentation and so on [19, 20]. The idea of FCM is to obtain the partitioned data and corresponding membership matrix by minimizing a objective function that is the sum of weighted absolute distances between cluster centers and each data point. It is noteworthy that the membership matrix would be utilized for later combination of the identified ARX models. The mathematical equation of objective function is depicted as follows.

$$J_{FCM} = \sum_{i=1}^{G}\sum_{j=1}^{D}(\mu_{j,i})^q||x_i - c_j||. \tag{3.4.6}$$

where x_i means the data point and G stands for the amount of data items. c_j denotes the center of clusters and the total number of clusters is D. $\mu_{j,i}$ expresses the membership matrix and it means the degree of x_i belonging to the jth cluster. q is defined as a scaling factor on each fuzzy membership. Moreover, some constraints should be considered within the minimization of objective function, which is described as following:

$$\begin{aligned}
&\mu_{j,i} \in [0,1], \quad for \quad 1 \le i \le G, 1 \le j \le D, \\
&\sum_{j=1}^{D}\mu_{j,i} = 1, \quad for \quad 1 \le i \le G, 1 \le j \le D, \\
&0 < \sum_{i=1}^{G}\mu_{j,i} < G, \quad for \quad 1 \le i \le G, 1 \le j \le D.
\end{aligned} \tag{3.4.7}$$

After setting the initial value of membership matrix $\mu_{j,i}$ and cluster centers c_j, the fuzzy clustering is carried out through a method of iterative optimization which is worked for updating the membership matrix and cluster centers. The equations

are shown as below:

$$\mu_{j,i} = \frac{1}{\sum_{k=1}^{D} (\frac{||x_i - c_j||}{||x_i - c_k||})^{2/(q-1)}}, \tag{3.4.8}$$

$$c_j = \frac{\sum_{i=1}^{G} (\mu_{j,i})^q x_i}{\sum_{i=1}^{G} (\mu_{g,i})^q}. \tag{3.4.9}$$

The terminal condition for iteration is $\max_{ij}|\mu_{j,i}(s+1) - \mu_{j,i}(s)| < \epsilon$ or $s < N_{set}$, where ϵ is defined as a termination criterion, s means the iterative step and N_{set} is the set point of terminal steps.

As the total sub-models may contribute to the future prediction, it is better to identify them simultaneously [21]. The detailed computing trick is represented as follows:

$$Y = [\theta_1 \quad \theta_2 \cdots \theta_D] \times [U_1 \odot X \quad U_2 \odot X \cdots U_D \odot X]^{\mathrm{T}}, \tag{3.4.10}$$

in which $\odot$ means Hadamard product, and U stands for an augment of membership matrix. The extended matrix is shown as below:

$$U_j = \begin{bmatrix} \mu_{j,1} & \mu_{j,1} & \cdots & \mu_{j,1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{j,G} & \mu_{j,G} & \cdots & \mu_{j,G} \end{bmatrix}_{G \times D}. \tag{3.4.11}$$

Then based on the operation introduced above, all the parameters θ and sub-models could be derived through PLS. In addition, to represent the total nonlinearities in DC system, all the sub-models should be combined into one equation through appropriate weights which are related to the membership matrix mentioned above.

The final integrated ARX model is shown as below:

$$x_k = \sum_{d=1}^{D} \omega_d [\sum_{i=1}^{N_x} \alpha_{i,d} x_{k-i} + \sum_{j=0}^{N_u} \beta_{j,d} u_{k-j} + \zeta_d \lambda + \sigma_d]. \tag{3.4.12}$$

Specifically, in this work x_k stands for the temperature in the rack, u_k denotes the control inputs including air flow rate and water flow rate, and ω_d means the weight of each sub-models. Besides, temperature of input water and workload of the whole rack are considered as measurable disturbances and defined as vector λ. ζ_g represents the coefficient vector of disturbances and σ_d is constant factor. It should be pointed out that modeling bias $b(k)$ is eliminated during the calculation of PLS. In summary, the proposed data-driven model has the following advantages. 1) It is suitable for the complex system which is difficult to model through first principles. 2) The multiple linear models used in this method are efficient to describe the nonlinear dynamics in the system. 3) The final form of ARX model could be applied for the controller design conveniently. The schematic diagram of developed data-driven modeling is described in Figure 3.4.

3.5 Fault tolerant predictive Control

To accurately regulate the temperature in DCs, many control strategies such as on-off control, PID control and MPC are employed. Particularly, MPC is widely used for DC management due to the ability of dealing with time series prediction, constrained optimization and nonlinear systems [22, 23, 24, 25, 26]. In our previous work [27], MPC control is also utilized to implement the temperature regulation based on data-driven model. However, during the real industrial operation, we found that some occasional

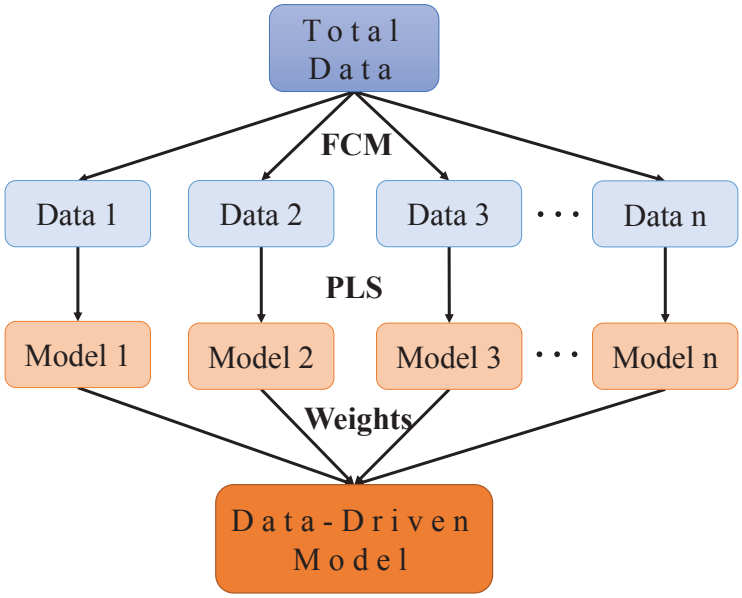

Figure 3.4: Schematic diagram of developed data-driven modeling.

actuator faults would deteriorate the performance of data-driven predictive controller and further threaten the security and reliability of entire system. Consequently, in order to mitigate the effects of actuator faults and ensure the safety of DCs, a novel fault tolerant part is considered in the MPC framework. At last, three experimental cases demonstrate the effectiveness and practicability of proposed controller.

3.5.1 Model predictive control

MPC is known as an advanced process control where the control actions are acquired by predicting the future states in a finite horizon and optimizing corresponding objective function iteratively [28]. Such predictive characteristic based on system model brings superior control performance [29]. Nevertheless, the ability of dealing with state and input constraints is another strength of MPC, and thus it is widely applied

for practical industry [30, 31, 32].

First, we transform the proposed data-driven model to a general discrete form (Based upon the validated experiments, N_x is selected as 1, N_u is set as 0 in the data-driven model):

$$x_k = Ax_{k-1} + Bu_k + d, \tag{3.5.1}$$

where x is temperature, $u = [F_a, F_w]^{\mathrm{T}}$ represents the vector of air flow rate and water flow rate, A and B are the coefficient, and d denotes the disturbance. The main challenge of MPC is to calculate the inputs sequences through constrained optimization of cost function at each sampling time. The procedure could be formulated as below:

$$\begin{aligned} \min \quad & J = \sum_{k=1}^{M} (x_k - x_{ref})^{\mathrm{T}} \Upsilon (x_k - x_{ref}) + \Delta u_k^{\mathrm{T}} \Phi \Delta u_k, \\ \text{s.t.} \quad & x_k = Ax_{k-1} + Bu_k + d, \\ & u_{k,min} \leq u_k \leq u_{k,max}, \\ & \Delta u_{k,min} \leq \Delta u_k \leq \Delta u_{k,max}, \end{aligned} \tag{3.5.2}$$

where J means the cost function, M is the prediction horizon, $\Delta u_k = u_k - u_{k-1}$ indicates the change of inputs in each step, x_{ref} is the reference value of state, Υ and Φ are weighting parameters, $u_{k,min}$ represents the lower bound of inputs and $u_{k,max}$ is the upper bound of inputs.

The optimizing procedure above is just for one step. At this time-step, once the optimization process yields a finite control sequence, the first control action in this sequence will be applied to the system. Then the same procedure would be repeated continuously with new measurements for the following sampling instant.

3.5.2 Fault tolerant control

As mentioned above, some actuator faults during the real industrial operation usually degrade the performance of our developed controller considerably. Therefore, fault tolerant control (FTC) is proposed in this work to compensate for the adverse influence induced by actuator faults, and further ensure the operational safety of D-C system. Basically, FTC could be categorized into active FTC and passive FTC [33, 34, 35]. These two methods have different design philosophies, but have the same control objective [36]. Active FTC indicates that the designed controller could reorganize the control structure online according to the system failures. Besides, a fault diagnosis scheme is necessary in the active FTC to offer the real-time faults of the system [37, 38]. On the contrary, passive FTC is able to dispose of the system faults without any faults information and control structure adjustments [39]. This kind of controller is fixed and insensitive to failures because of the pre-designed redundancy and robustness, and thus it is named as passive FTC.

The actuators in this system are direct current fans and electrical motorized ball valve, which are utilized to control the air flow rate and water flow rate respectively. Based on the feedbacks from engineers, there are three common actuator faults including fans aging, fans failure and valve deactivation. Generally, it is difficult to deal with three different actuator faults simultaneously in a fixed control structure, therefore, active FTC is employed in this work instead of passive FTC. The detailed methodologies are shown as following.

Fault model

In order to model the actuator faults, a fault factor is proposed in the previous data-driven model:

$$x_k = Ax_{k-1} + B\rho_k u_k + d, \tag{3.5.3}$$

where $\rho = diag(\rho_a, \rho_w)$ is the fault factor vector which denotes the ratios between the real actuator responses and control commands from controller. If the actuator responses are equal to the control commands, it means the system is healthy and $\rho = I$; if the actuators lose functionality completely $\rho = 0$.

Fault identification

To compensate for the faults and work out appropriate control actions in real time, it is essential to do the fault identification. The detailed procedure of fault identification in this article is online calculation or estimation of fault factor. The formula of fault factor is presented as below:

$$\rho_i = \begin{cases} \frac{\nu_{act}}{\nu_{com}}, & \text{if } |\nu_{act} - \nu_{com}| \geq \delta \\ 1, & \text{if } |\nu_{act} - \nu_{com}| \leq \delta, \end{cases} \tag{3.5.4}$$

in which ν_{act} means the real actuator action, and ν_{com} is defined as the control command. δ represents the threshold for fault diagnosis which is determined by repeated experiments. In general cases of fault identification, the real actuator information are usually estimated through input observers such as Kalman filter, sliding mode observer, moving horizon estimation and so on. However, for real industrial applications, the customers have high demands of information accuracy and reliability, especially

the input and output data. Consequently, several physical sensors are employed to measure the actuator actions in this program. Furthermore, for the measurement of fan speed, it is quite convenient to use the fan's own internal hall effect sensor. The water flow rate is measured through the system-provided flow meter. These two approaches of obtaining actuators information are all more reliable than input observers without increasing costs.

As we can see in Figure 3.3, there are five fans in the cooling unit. Thus we need to identify five fault factors of fans at the same time and combine them together. The simple equation of fault factors is shown as follows.

$$\rho = \begin{bmatrix} \rho_a \\ \rho_w \end{bmatrix} = \begin{bmatrix} 1/5(\rho_{a,1} + \rho_{a,2} + \rho_{a,3} + \rho_{a,4} + \rho_{a,5}) \\ \rho_w \end{bmatrix}. \tag{3.5.5}$$

Here, for the healthy system, all the fault factors of fans $\rho_{a,i} = 1$, and then the final fault factors of fans $\rho_a = 1$. For one of the fans fails to work, the final fault factors of fans $\rho_a = 0.8$ and it means the performance of airflow actuator decreases by 20%.

Fault tolerant MPC

The fault factor would be identified in each sampling instant, and the state-space model is updated accordingly. Consequently, the problem of fault tolerant MPC

could be reformulated as below:

$$\begin{aligned} \min \quad & J^* = \sum_{k=1}^{M} (x_k - x_{ref})^{\mathrm{T}} \Upsilon (x_k - x_{ref}) + \Delta u_k^{\mathrm{T}} \Phi \Delta u_k, \\ \text{s.t.} \quad & x_k = A x_{k-1} + B \rho_k u_k + d, \\ & u_{k,min} \leq u_k \leq u_{k,max}, \\ & \Delta u_{k,min} \leq \Delta u_k \leq \Delta u_{k,max}. \end{aligned} \tag{3.5.6}$$

Additionally, it is also noteworthy that the fault of valve deactivation would result in the change of cost function in fault tolerant MPC. When the fault of valve deactivation occurs, the water flow rate stays as a constant and cannot be controlled any more. Thus the MISO system changes to a single-input-single-output (SISO) system, and the cost function should be adjusted accordingly. The updated fault tolerant MPC problem is represented as:

$$\begin{aligned} \min \quad & J^{\#} = \sum_{k=1}^{M} (x_k - x_{ref})^{\mathrm{T}} \Upsilon (x_k - x_{ref}) + \Delta u_k^{*\mathrm{T}} \Phi \Delta u_k^*, \\ \text{s.t.} \quad & x_k = A x_{k-1} + B \rho_k u_k^* + d, \\ & u_{k,min}^* \leq u_k^* \leq u_{k,max}^*, \\ & \Delta u_{k,min}^* \leq \Delta u_k^* \leq \Delta u_{k,max}^*, \end{aligned} \tag{3.5.7}$$

where $u_k^* = [F_a, \nu_{act,w}]^{\mathrm{T}}$, and $\nu_{act,w}$ is the constant measurement of water flow rate under valve deactivation and no longer a control input. Finally, by applying a switch trigger and this updated cost function, the fault of valve deactivation is addressed. To clearly represent the procedure of fault tolerant MPC in this DC system, a flowchart is provided in Figure 3.5.

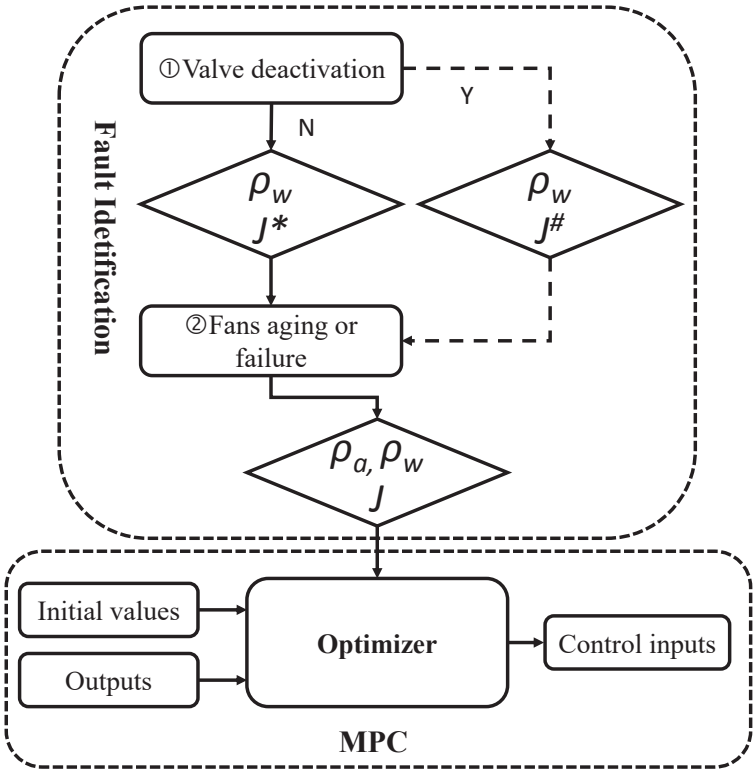

Figure 3.5: Flowchart of the fault tolerant MPC algorithm in this work.

3.6 Experimental results

All the experiments are implemented in a real single-rack DC with rack-based cooling architecture which is shown in Figure 3.6. The experimental DC system comprises 30 servers, cooling unit, several sensors, pipes and chilled water supply system from the building. The control algorithm is programmed in Python environment and conducted through Raspberry Pi Zero which is regarded as the master micro-controller in this work. Besides, considering reducing the computational burden of Raspberry Pi, an Arduino board (Nano) is used as the slave micro-controller whose function is to send actuators commands of Raspberry Pi and collect data from sensors. Additionally, the

IT workload in this experiment is set as a constant value.

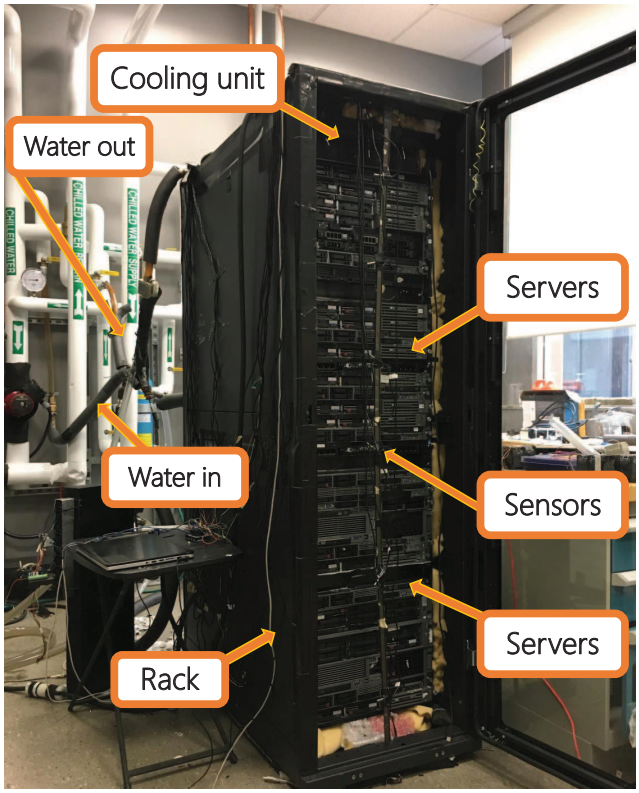

Figure 3.6: Experimental DC with rack-based cooling architecture.

In order to obtain an efficient data-driven model, the experiment under all working conditions is carried out and the relevant measurements are utilized for model training. The experiment costs nearly 80 hours and includes 10 cycles covering different air flow, water flow and IT workload. The experimental cycle is represented in Figure 3.7. Then a group of data from another experiment is used to validate the performance of data-driven model, and the result of validation is depicted in Figure 3.8. As we can see in this figure, the error of proposed model is roughly around 0.2 degree, which is within the acceptable boundary and further demonstrates the data-driven model is able to describe the DC system.

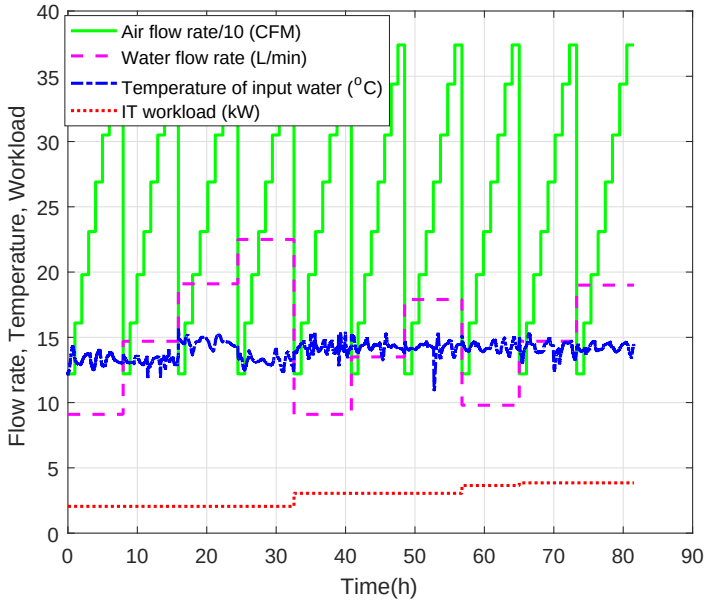

Figure 3.7: Measurements under all working conditions for model training.

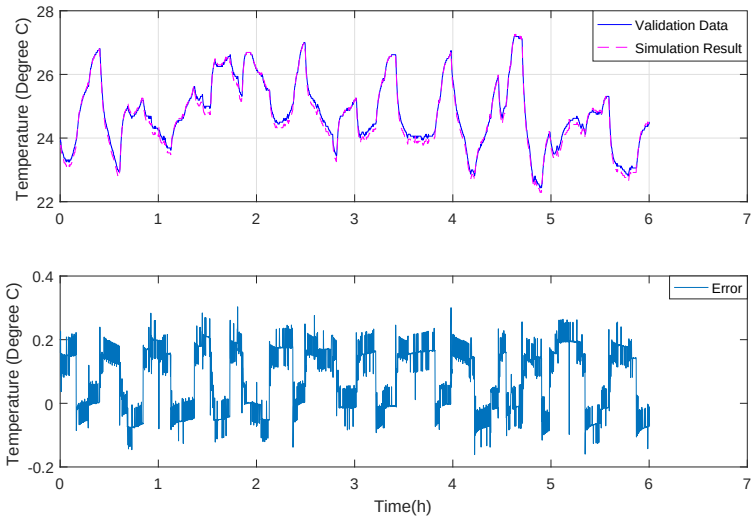

Figure 3.8: Validation result of data-driven model.

As mentioned above, three kinds of actuator faults comprising aging of fans, failure of fans and valve deactivation are investigated in this paper. Therefore, three groups

of experiment concerning the actuator faults are carried out to test and verify the effectiveness of designed controller. The detailed experiment results are presented as follows.

3.6.1 Case I: Aging of fans

The first type of actuator faults is aging of fans, which indicates the performance of total fans degrade by a certain percent over time. In this experiment, the fault is injected at 400s and the fans' performance affected by fault is assumed to degrade by 30 percent. The comparison of results between data-driven predictive controller with fault tolerant control and without fault tolerant control are depicted in Figure 3.9. Moreover, the control inputs and measured disturbance are shown in Figure 3.10 as well.

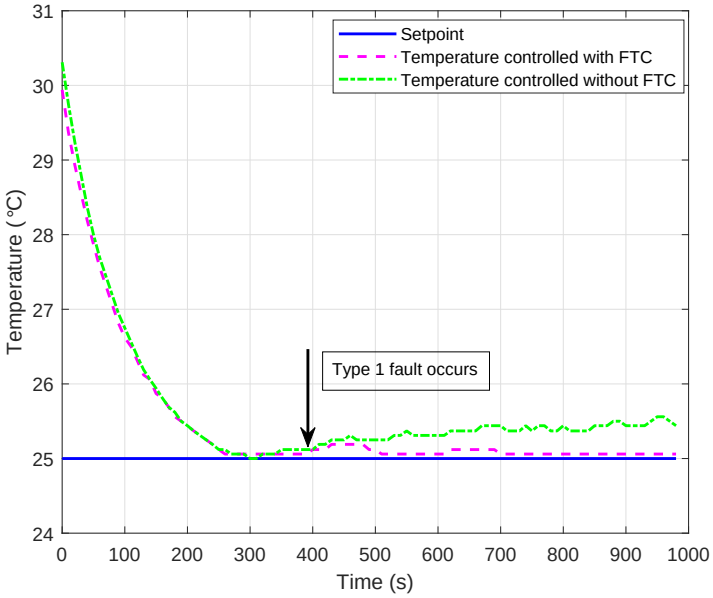

Figure 3.9: The comparative performance of data-driven predictive controller with fault tolerant control and without fault tolerant control in case I.

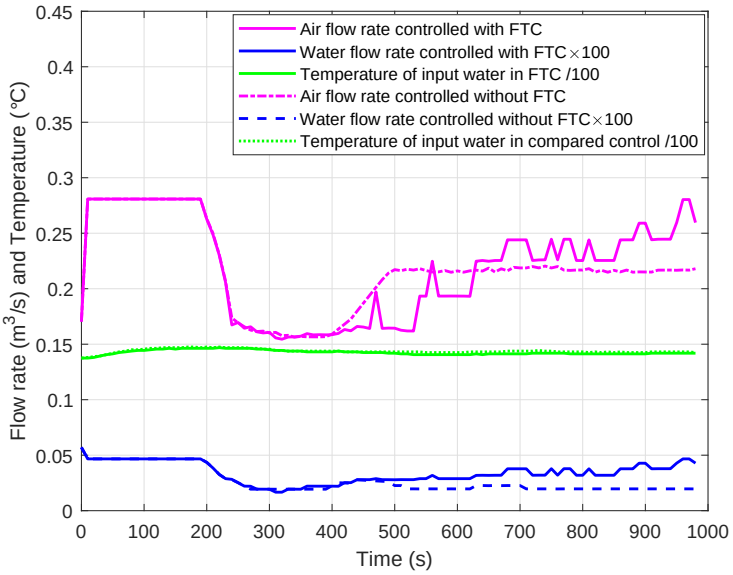

Figure 3.10: Control inputs and measured disturbances in case I.

As seen in Figure 3.9, temperatures regulated by both of the controllers are able to track the set point properly before fault occurs, which represents the good predictive capability of data-driven model and control accuracy of developed predictive controller. When fault is injected at 400s, the data-driven predictive controller without fault tolerant control perform badly and the temperature in the rack keeps rising. On the contrary, the proposed controller with fault tolerant control works properly under fault and the temperature could still follow the set point well. Besides, the control inputs after fault occurrence described in Figure 3.10 also prove the superiority of fault tolerant predictive control.

3.6.2 Case II: Failure of fans

The second type of actuator faults means that one or some of the fans stop working suddenly in cooling unit, which is one of the major faults in fans. As the other fans

still work normally under this situation, the temperature in the system could be regulated by some proper control algorithms. In this case, we assume that one of the fans stops working after the system reaches the steady state and the fault is injected at a specific point-in-time through the code in advance. The comparative experiments of data-driven predictive controller with fault tolerant control and without fault tolerant control are conducted respectively and the results are described in Figure 3.11. Besides, the control inputs are represented in Figure 3.12 as well.

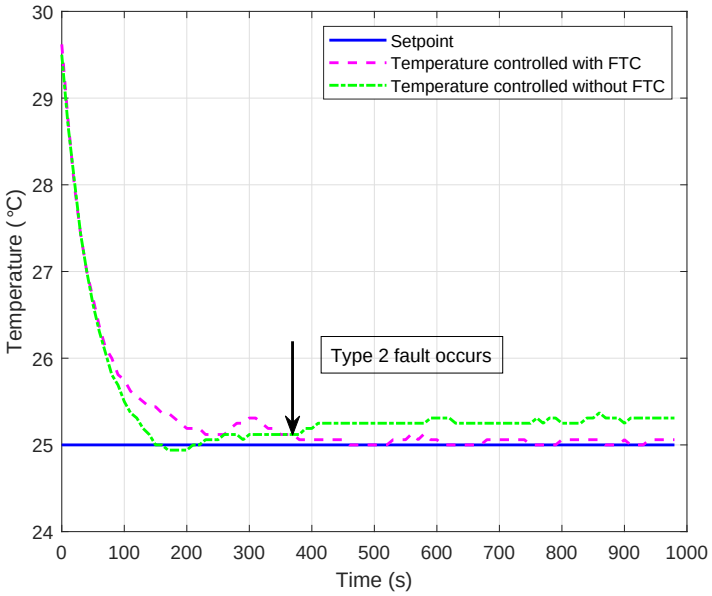

Figure 3.11: The comparative performance of data-driven predictive controller with fault tolerant control and without fault tolerant control in case II.

In Figure 3.11, the effectiveness of controller with fault tolerant control is satisfying after the injection of fault, while the controller without fault tolerant control shows a bad adaptability to the actuator fault. Furthermore, the differences of control commands shown in Figure 3.12 express the internal reason why the controller with fault tolerant control performs better. It is also noteworthy that the control

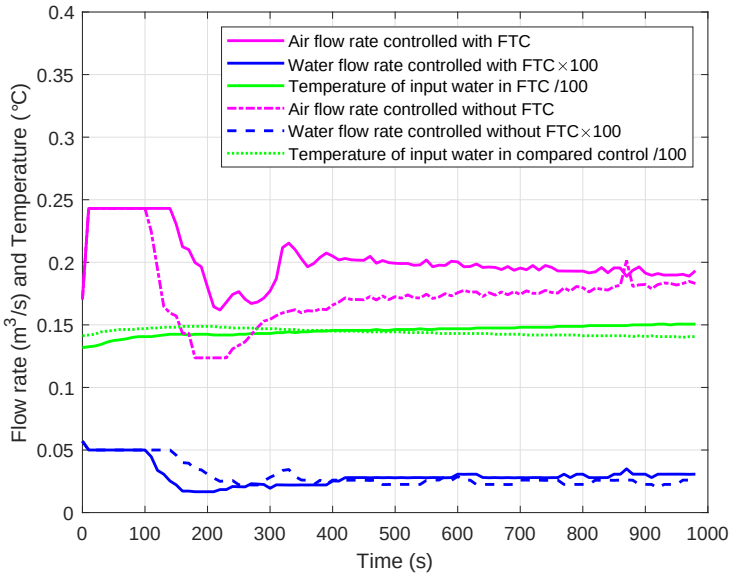

Figure 3.12: Control inputs and measured disturbances in case II.

performance is quite sensitive to the change of input water temperature and the temperature fluctuates greatly at the beginning period, thus the controlled temperatures in the rack overshoot and oscillate around the set point.

3.6.3 Case III: Valve deactivation

The third common actuator fault in cooling unit is the valve deactivation. The valve fault is also injected around 400s and the pre-arranged stopped water flow rate is set as 0.000333 m^3/s or $20L/min$ (it can also be set as another value). The comparison of effects between two controllers is presented in Figure 3.13 and corresponding control commands are shown in Figure 3.14.

As the Figure 3.13 describes, the temperatures could follow the set point precisely, and there aren't obvious overshoot and undershoot before the the injection of fault. When the valve fault occurs, the temperature managed through controller

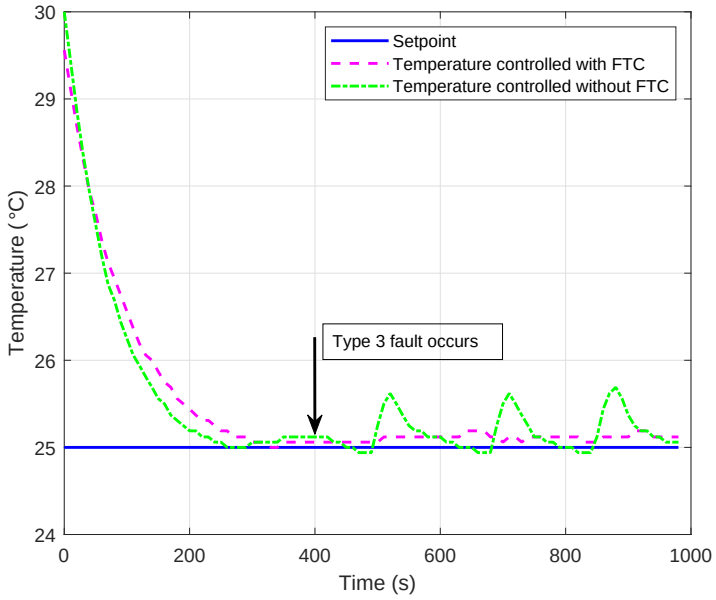

Figure 3.13: The comparative performance of data-driven predictive controller with fault tolerant control and without fault tolerant control in case III.

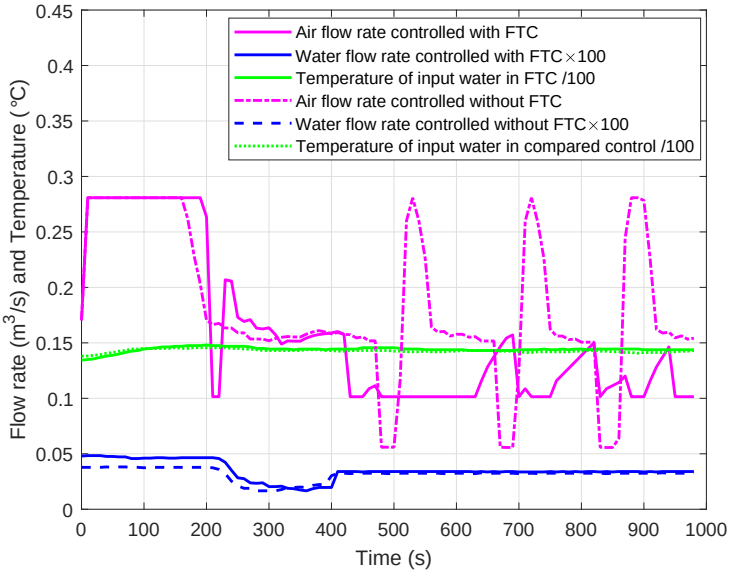

Figure 3.14: Control inputs and measured disturbances in case III.

without fault tolerant control appears a big oscillation due to the change of control structure. In addition, the control input of controller without fault tolerant control varies significantly as well. However, the behavior of controller with fault tolerant control is absolutely excellent even under the occurrence of fault owing to the timely adjustment of cost function in control algorithm.

As we can see in the three cases of experiments, the data-driven fault tolerant predictive controller behaves quite well in the respects of response time, fault tolerance, overshoot and undershoot. Besides, the comparison of predictive controller without fault tolerant control further demonstrate the remarkable performance of developed controller.

3.7 Conclusion

This paper presents a novel data-driven fault tolerant predictive controller for temperature regulation of DC with rack-based cooling architecture. First, a data-driven based on combined ARX models is established, which is identified through PLS and FCM. Then, according to such data-driven model, fault tolerant predictive control is implemented. In this work, this common actuator faults containing fan aging, fan failure and valve deactivation are considered to validate the controller. Finally, the comparative results from real experiments reveal that the proposed controller works properly under different fault cases. In the future work, more approaches of data-driven modeling for industrial applications and strategies for power-efficient optimization under temperature constraint in DC will be explored.

Bibliography

[1] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renew. Sustain Energy Rev.*, vol. 31, pp. 622–638, Mar. 2014.

[2] H. Cheung, S. Wang, C. Zhuang, and J. Gu, "A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation," *Appl. energy*, vol. 222, pp. 329–342, Jul. 2018.

[3] H. Moazamigoodarzi, P. J. Tsai, S. Pal, S. Ghosh, and I. K. Puri, "Influence of cooling architecture on data center power consumption," *Energy*, vol. 183, pp. 525–535, Sep. 2019.

[4] T. Lu, X. Lü, M. Remes, and M. Viljanen, "Investigation of air management and energy performance in a data center in finland: Case study," *Energy and Buildings*, vol. 43, no. 12, pp. 3360–3372, Dec. 2011.

[5] H. Moazamigoodarzi, S. Pal, S. Ghosh, and I. K. Puri, "Real-time temperature predictions in IT server enclosures," *Int. J. Heat Mass Transf.*, vol. 127, pp.

890–900, Dec. 2018.

[6] T. D. Boucher, D. M. Auslander, C. E. Bash, C. C. Federspiel, and C. D. Patel, "Viability of dynamic cooling control in a data center environment," *J. Electron. Packag.*, vol. 128, no. 2, pp. 137–144, Nov. 2006.

[7] H. Zhang, S. Shao, H. Xu, H. Zou, and C. Tian, "Free cooling of data centers: A review," *Renew. Sustain Energy Rev.*, vol. 35, pp. 171–182, Jul. 2014.

[8] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. Peng, "Green data center with IoT sensing and cloud-assisted smart temperature control system," *Computer Networks*, vol. 101, pp. 104–112, Jun. 2016.

[9] L. Parolini, B. Sinopoli, B. H. Krogh, and Z. Wang, "A cyber–physical systems approach to data center modeling and control for energy efficiency," *Proc. IEEE*, vol. 100, no. 1, pp. 254–268, Jan. 2012.

[10] B. Durand-Estebe, C. Le Bot, J. N. Mancos, and E. Arquis, "Data center optimization using PID regulation in CFD simulations," *Energy and Buildings*, vol. 66, pp. 154–164, Nov. 2013.

[11] J. Deng, L. Yang, X. Cheng, and W. Liu, "Self-tuning PID-type fuzzy adaptive control for CRAC in datacenters," in *International Conference on Computer and Computing Technologies in Agriculture*, pp. 215–225, 2013.

[12] R. Zhou, Z. Wang, C. E. Bash, and A. McReynolds, "Modeling and control for cooling management of data centers with hot aisle containment," in *Proceedings of ASME 2011 International Mechanical Engineering Congress & Exposition*, 2011.

[13] M. Ogawa, H. Fukuda, H. Kodama, H. Endo, T. Sugimoto, T. Kasajima, and M. Kondo, "Development of a cooling control system for data centers utilizing indirect fresh air based on model predictive control," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 132–137, 2015.

[14] L. Parolini, B. Sinopoli, and B. H. Krogh, "Model predictive control of data centers in the smart grid scenario," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 10 505–10 510, Jan. 2011.

[15] T. Ding, Z. guang He, T. Hao, and Z. Li, "Application of separated heat pipe system in data center cooling," *Appl. Therm. Eng.*, vol. 109, pp. 207–216, Oct. 2016.

[16] B. Corbett and P. Mhaskar, "Subspace identification for data-driven modeling and quality control of batch processes," *AIChE J.*, vol. 62, no. 5, pp. 1581–1601, Jan. 2016.

[17] S. Aumi and P. Mhaskar, "Integrating data-based modeling and nonlinear control tools for batch process control," *AIChE journal*, vol. 58, no. 7, pp. 2105–2119, Jul. 2012.

[18] S. Aumi and P. Mhaskar, "An adaptive data-based modeling approach for predictive control of batch systems," *Chem. Eng. Sci.*, vol. 91, pp. 11–21, Mar. 2013.

[19] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, Aug. 2005.

[20] D. Jin, X. Bai, and Y. Wang, "Integrating structural symmetry and local homoplasy information in intuitionistic fuzzy clustering for infrared pedestrian segmentation," *IEEE Trans. Syst., Man, Cybern., Syst.*, 2019.

[21] A. Siam, C. Brandon, and M. Prashant, "Data-based modeling and control of nylon-6, 6 batch polymerization," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 1, pp. 94–106, Jan. 2013.

[22] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, Dec. 2014.

[23] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[24] M. Farina, L. Giulioni, and R. Scattolini, "Stochastic linear model predictive control with chance constraints–a review," *J. Process Control*, vol. 44, pp. 53–67, Aug. 2016.

[25] T. Gao, S. Yin, J. Qiu, H. Gao, and O. Kaynak, "A partial least squares aided intelligent model predictive control approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 11, pp. 2013–2021, Nov. 2018.

[26] M. Cannon, "Efficient nonlinear model predictive control algorithms," *Annu. Rev. Control*, vol. 28, no. 2, pp. 229–237, Jan. 2004.

[27] S. Shi, "Rack-based data center temperature regulation using data-driven model predictive control," Master's thesis, McMaster University, 2019.

[28] D. Laurí, J. A. Rossiter, J. Sanchis, and M. Martínez, "Data-driven latent-variable model-based predictive control for continuous processes," *J. Process Control*, vol. 20, no. 10, pp. 1207–1219, Dec. 2010.

[29] D. Piga, M. Forgione, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven MPC design," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 577–582, Jan. 2019.

[30] M. A. Evans, M. Cannon, and B. Kouvaritakis, "Robust MPC tower damping for variable speed wind turbines," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 290–296, Jul. 2014.

[31] B. Sakhdari and N. L. Azad, "Adaptive tube-based nonlinear MPC for economic autonomous cruise control of plug-in hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 11 390–11 401, Dec. 2018.

[32] G. P. Incremona, A. Ferrara, and L. Magni, "MPC for robot manipulators with integral sliding modes generation," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1299–1307, Jun. 2017.

[33] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annu. Rev. Control*, vol. 32, no. 2, pp. 229–252, Dec. 2008.

[34] S. Yin, H. Luo, and S. X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, May. 2013.

[35] J. Lan and R. J. Patton, "A new strategy for integration of fault estimation within fault-tolerant control," *Automatica*, vol. 69, pp. 48–59, Jul. 2016.

[36] J. Jiang and X. Yu, "Fault-tolerant control systems: A comparative study between active and passive approaches," *Annu. Rev. Control*, vol. 36, no. 1, pp. 60–72, Apr. 2012.

[37] R. Wang and J. Wang, "Fault-tolerant control with active fault diagnosis for four-wheel independently driven electric ground vehicles," *IEEE Trans. Veh. Technol.*, vol. 60, no. 9, pp. 4276–4287, Nov. 2011.

[38] J. Wang, J. Zhang, B. Qu, H. Wu, and J. Zhou, "Unified architecture of active fault detection and partial active fault-tolerant control for incipient faults," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1688–1700, Jul. 2017.

[39] H. Niemann and J. Stoustrup, "Passive fault tolerant control of a double inverted pendulum-a case study," *Control Eng. Pract.*, vol. 13, no. 8, pp. 1047–1059, Aug. 2005.

Chapter 4

Thermal Modeling in Data Center via Input Convex Neural Networks overcoming Catastrophic Forgetting

This chapter is reproduced from *"Thermal Modeling in Data Center via Input Convex Neural Networks overcoming Catastrophic Forgetting", **Kai Jiang**, and Fengjun Yan, submitted to Mechatronics, 2020.* The author of this thesis is the first author and the main contributor of this publication.

4.1 Abstract

Currently increasing number of data centers (DCs) are built to support the development of information technology (IT). In DCs, thermal dynamics play an significant

role in security, reliability and efficiency. Thus this paper emphatically investigates the thermal modeling in DCs through deep neural networks (DNNs). As sometimes the developed thermal model is not only dedicated to temperature prediction but also used for real-time control, the control tractability of neural network should be taken into consideration. In this work, a novel model named input convex neural network (ICNN) is adopted, which is trained under constrained weights, and convex with respect to the inputs. By applying such ICNN, the optimization problem in control is guaranteed to be solvable. Furthermore, to overcome the problem of catastrophic forgetting during sequential training, the approach of elastic weight consolidation (EWC) is adopted. Finally, model validations based on real experimental data are implemented, and the results demonstrate the developed model is remarkable.

Key words: Thermal modeling, Deep Neural Networks, Catastrophic forgetting, Data center.

4.2 Introduction

Nowadays a series of DCs are established to satisfy the increasing demands of information and communication services [1, 2, 3]. Meanwhile, appropriate approaches of thermal management are required to guarantee the safe operation of IT equipments (ITEs) such as servers and storage devices in DCs. As is known, temperature monitoring is an important branch of thermal management in DCs [4]. In general, numbers of physical sensors are mounted everywhere to monitor the real-time temperature in the plant. However, when the thermal anomalies in DCs occur, the equipments have been damaged irreversibly already. Therefore, a temperature prediction model is usually essential to predict the thermal dynamics in advance, and further provide us more

time to deal with the problems [5, 6, 7].

Temperature model also makes an important contribution to the control strategy of cooling infrastructure, as the control strategy is developed to generate optimal control actions by minimizing the errors between model-based temperature predictions and reference temperatures. The controller in cooling system is mainly utilized to regulate the environmental temperature in the facility and thus ensure the ITEs work normally [8, 9, 10, 11]. Nevertheless, an effective control strategy of cooling system is also crucial to the energy efficiency of DCs. Cooling system usually consumes large amounts of energy, which accounts for nearly forty percent of the total power consumption in a DC [12, 13]. The controller is able to make proper allocation of cooling resources according to the temperature distribution, whereby avoiding overcooling and energy waste [14, 15]. Consequently, a temperature prediction model with control tractability and high accuracy is really necessary for the thermal management in DC.

In response to the demand for thermal modeling, many studies have been carried out. First, several physics models based on the thermodynamics and energy conservation principle are proposed. A thermal model was developed to predict the temperature distribution of DC in [16]. This paper mainly focused on fast physics modeling that describes the transient thermal behaviors. In [17], an adaptive numerical temperature model was presented to capture the temperature behaviors in DCs. This model was built under the framework of state-space model and dedicated to controller design. A computational fluid dynamics (CFD) model was studied to predict the thermal field for a small DC in [18]. The thermal boundary condition was particularly considered in this work, which reduced the prediction error greatly.

Besides, Hosein et al. investigated real-time temperature modeling in a DC with rack mountable cooling unit (RMCU) [19]. In this paper, a novel transient zonal model was proposed based upon mass and energy conservation. According to these articles, physics models are efficient to describe the internal thermal behaviors and predict the temperature. However, the physics models heavily rely on modeling assumptions, which severely degrade the prediction accuracy and result in performance deterioration. Moreover, due to the complicated heat transfer and air flow in DCs, the thermal modeling based on first principles is also a time-consuming task.

Compared with physics models, data-driven models are more widely used for thermal modeling in DCs [20]. In [21, 22], the approach of proper orthogonal decomposition (POD) was employed to capture the relationships between temperature and major variables. Kai et al. studied improved partial least square (PLS) to present the thermal behaviors in a DC, and combined with adaptive Kalman filter to estimate the temperature distribution [23]. The statistical modeling frameworks show their superiority in temperature prediction, however, the poor extrapolative accuracy restricts the further application. Neural network is another effective method to implement temperature prediction in DCs. The authors in [24, 25, 26] utilized conventional artificial neural networks (ANN) or convolutional neural network (CNN) to predict the temperature distribution in DCs. These works show that neural networks are capable of capturing the thermal dynamics in complex systems and predict the temperature precisely, while most of them fail to pay much attention to the control tractability of neural networks.

As we all know, DNNs are widely utilized for industrial systems owing to the strong representation capability [27]. However, they are seldom used for the following

optimal control or decision making due to the feature of nonlinearity and nonconvexity. To bridge the gap between high model accuracy and control tractability of DNNs, we adopt an ICNN modeling method [28, 29]. ICNN is constructed within the architecture of traditional neural networks, while the layer parameters are restricted to be non-negative to maintain the convexity. By using such ICNN, the issue of computational optimization in DNN-based optimal control could be converted to a convex optimization problem and solved easily.

Besides, in order to deal with the problem of catastrophic forgetting in multiple tasks learning, an algorithm of EWC is employed in this work [30]. This algorithm could figure out the important weights for previous tasks and slow down the changing of them to avoid forgetting old tasks. Then the other unimportant weights would be updated to fit the new tasks. Finally, the simulations according to the data from different tasks present the effectiveness of our proposed algorithm in continual learning. In addition, we also investigate the capability of this algorithm against noises in training data, and the results demonstrate its impressive performance.

The rest of this article is organized as following. Section II and III present the configuration of DC and the methodology of input convex neural networks separately. The detailed experimental results in simulation platform are provided in Section IV. Finally, conclusion is depicted in V.

4.3 Fundamental of DC with rack-based cooling architecture

Cooling system plays an significant role in DC to regulate temperature and further protect servers. Currently, there are three types of cooling methods in DCs, which contains air cooling, liquid cooling and phase-change-material cooling [31]. Considering the operational feasibility, cooling security and equipment cost, the method of air cooling is widely adopted in DC cooling system [32]. In light of cooling architecture, room-based, row-based and rack-based cooling architecture are regarded as three typical architectures. Room-based cooling architecture means that the cooling system is mounted in a room and provides cold air to the whole room to cool the devices in DCs directly. Row-based cooling architecture indicates that the cooling unit is installed in a row of connected racks and used for cooling the devices inside. Rack-based cooling architecture is designed to cool the devices in each enclosed individual rack [33, 34]. The configurations of three type of cooling architecture are presented in Figure 4.1.

According to the studies in [35], row-based and rack-based cooling system are more energy-efficient over room-based cooling system owing to the shorter airflow path and better airflow distribution. Besides, in row-based and rack-based cooling systems, rack-based cooling system is more suitable to DCs with stand-alone high-density racks due to the flexibility and scalability. Thus, the detailed investigation relating to rack-based cooling architecture is conducted in this work.

The main concept of rack-based cooling system in DCs is to extract the heat generated by servers out of rack through the heat exchange between chilled water and hot air. The airflow from hot aisle to cold aisle is driven by several fans and

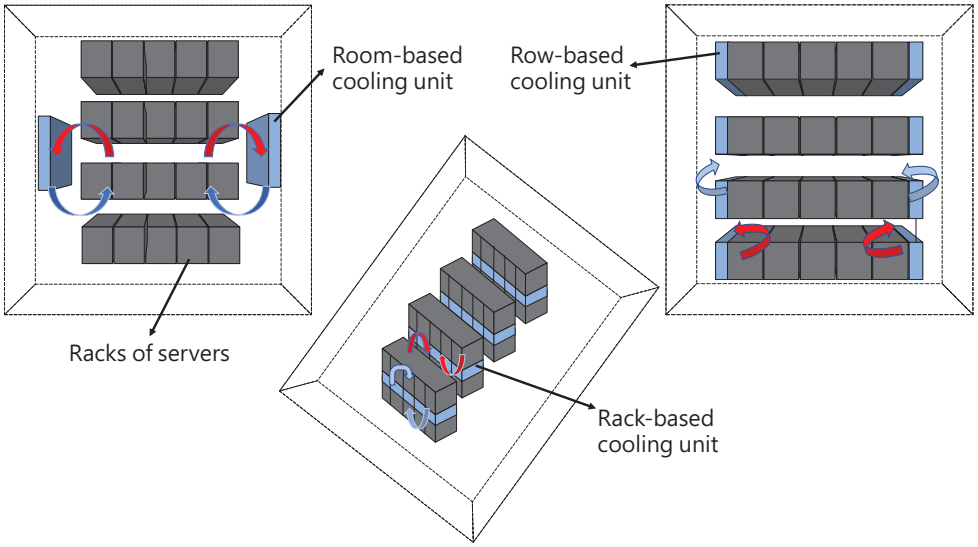

Figure 4.1: Configurations of DCs with room-based, row-based and rack-based cooling architecture

.

goes through a heat exchanger filled with circulating chilled water, then the heat of hot air would be taken away via the chilled water. The circulating water is supplied and cooled through a chiller outside. The whole system is depicted in Figure 4.2. The RMCU is also represented in Figure 4.3, which comprises fans, heat exchanger, valve, micro controller, cables, power source and some pipes. Moreover, the location of RMCU in the rack could be adjusted based on the demand of customers.

As is shown in Figure 4.2 and Figure 4.3, we can figure out the thermal control system in DC clearly. The output or the control objective is the temperature T in the front of rack. The inputs are the air flow rate F_a controlled by fans and the water flow rate F_w controlled by valve. In addition, the temperature of inlet water T_w and IT workload Θ are considered as measured disturbances, which are determined by the chiller system in the building and network traffic respectively. The discrete-time

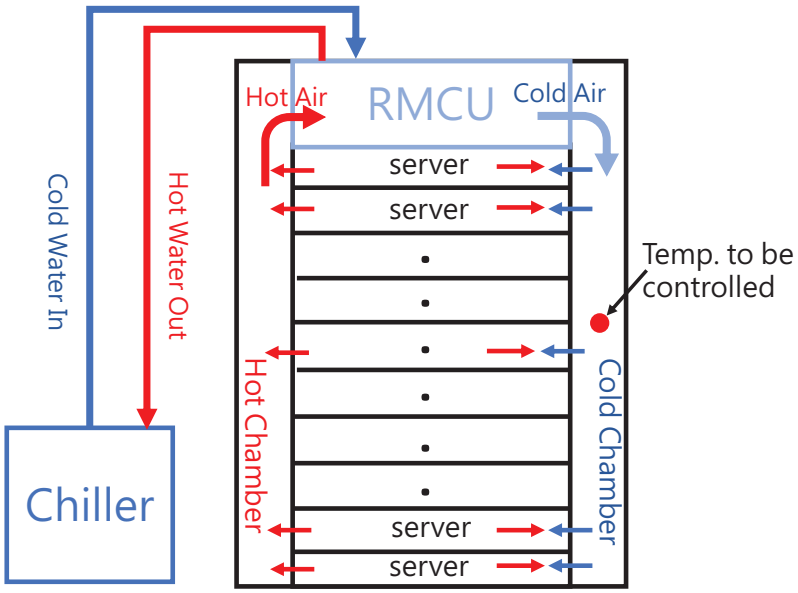

Figure 4.2: Detailed configuration of DC with rack-based cooling architecture.

system model could be described as below:

$$T_t = f(T_{t-1}, U_t, R_t), \tag{4.3.1}$$

where

$$U_t = [F_{a,t}, F_{w,t}], \tag{4.3.2}$$

$$R_t = [T_{w,t}, \Theta_t]. \tag{4.3.3}$$

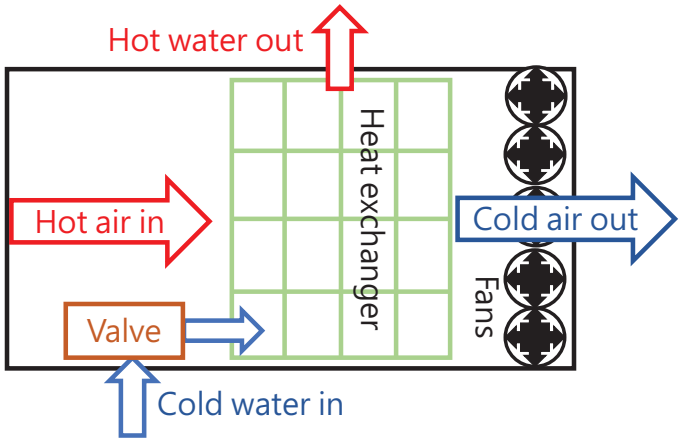

Figure 4.3: Internal structure of rack mountable cooling unit.

4.4 Modeling algorithm of input convex neural networks and elastic weight consolidation

4.4.1 Input convex neural networks

DNNs are widely applied to describe the nonlinear behaviors in industrial systems, while they are seldom used for dynamic modeling in control issues owing to their nonconvexities and nonlinearities [36]. Although DNNs are utilized for model-free or end-to-end control such as reinforcement learning sometimes, the online training of end-to-end control is really risky and time-consuming for real industrial systems [28]. To this end, the ICNN is proposed to bridge the gap between DNNs and optimal control. The ICNN is designed to be convex with regard to inputs through restricting the layer parameters to non-negative values. Then by using the ICNN, the calculation of control inputs in optimal control is guaranteed to be a convex optimization problem.

The architecture used in ICNN is recurrent neural network (RNN), therefore the proposed ICNN can also be called input convex recurrent neural network (ICRNN).

RNN is one of the most common neural networks in artificial intelligence, and many novel networks derived from RNN are utilized to deal with different tasks such as natural language processing (NLP), sequence classification, regression prediction and so on [37]. The most important characteristic of RNN is the recurrent connection on hidden layers which indicates the hidden neurons not only receive the current information from upper layer but also use the neural memory from previous step. Such feature enables RNN to have a short-term memory, and be applicable to sequence prediction problems [38]. Accordingly, RNN is adopted as the model structure to describe the thermal dynamics in our work. The detailed architecture of typical RNN is shown in Figure 4.4, and the corresponding equations are expressed as following:

$$h_{t,i} = \sigma_h(W_i h_{t-1,i} + C_i h_{t,i-1} + b_{t,i}), \tag{4.4.1}$$

$$y_t = \sigma_y(C_k h_{t,k} + b_{t,k}), \tag{4.4.2}$$

where h_t denotes the hidden neurons, u_t stands for the input, y_t means the output, and σ illustrates the activation function. W and C represent the weights of recurrent neurons and next layer respectively, b is the layer bias, and k indicates the number of hidden layers. Additionally, E in Figure 4.4 and Figure 4.5 is the weights of input layer.

As mentioned above, constraining the layer parameters within non-negative domain could ensure the neural network convex. This statement could be concluded as the following proposition:

Proposition 1. Any designed neural network is guaranteed to be convex from inputs

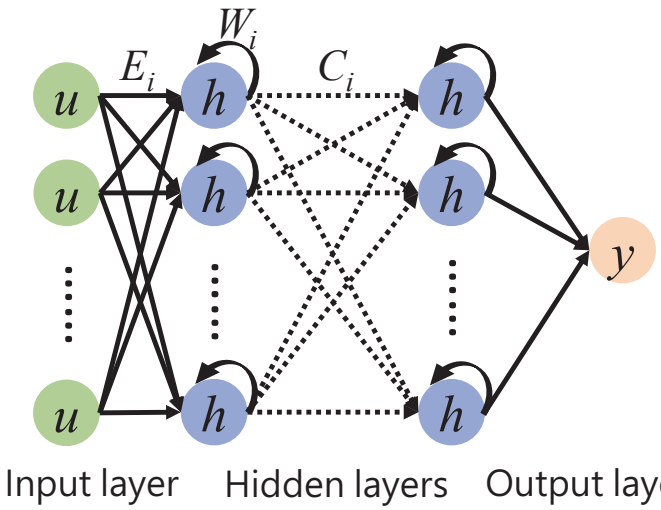

Figure 4.4: The detailed architecture of traditional RNN.

to output if all weights on hidden layers C and W are non-negative, and all activation functions σ should be convex and nondecreasing.

The proof of this proposition is quite simple, which could be proved through the following facts in [29, 28, 39]:

1) Affine function is still convex when its domain is a convex set.

2) Non-negative weighted sums of convex functions are still convex.

3) The composition of a convex function and convex non-decreasing function is still convex.

As is shown in Eq. (4.4.1), the $Wh+b$ and $Ch+b$ are all affine functions and should be convex when the domain is a convex set. Then, the non-negative weighted sum of these two affine functions could be proved convex based on the second condition. The composition of these two affine functions and convex non-decreasing activation function is proved convex as well according to the third condition. Besides, the Eq. (4.4.2) could be proved convex through the same method. Finally, the whole neural

network is guaranteed convex by constraining layer weights and activation functions. Actually, the constraint of activation functions is not restrictive, since many activation functions such as "rectified linear unit (ReLU)" or "maxpooling unit" already meet the requirement.

Obviously, the constraints in the proposition would degrade the representation capability of RNN, as the non-negative weights restrict the use of hidden layers and lead to part of information lost. To address this problem, the "passthrough" layers are proposed to improve the representation capability of ICRNN. "passthrough" layers demonstrate that the inputs u could be connected with each hidden layer directly, and thus compensate the information lost caused by weights constraints [40, 41]. The developed architecture of ICRNN with respect to "passthrough" layers is shown in Figure 4.5. Additionally, the weights on "passthrough" layers maintain non-negative as well, and thus the introduction of "passthrough" layers would not effect the convexity of ICRNN. The equations of ICRNN is accordingly transformed as follows:

$$h_{t,i} = \sigma_i(W_i h_{t-1,i} + C_i h_{t,i-1} + P_i u_t + b_{t,i}), \tag{4.4.3}$$

$$y_t = \sigma_k(C_k h_{t,k} + P_k u_t + b_{t,k}), \tag{4.4.4}$$

where P stands for the weights of developed "passthrough" layers.

The training process of ICRNN is similar to the traditional neural networks. The approach used for optimization in the training procedure is stochastic gradient descent (SGD). The only difference between the training of ICRNN and traditional neural networks is the constraints of final layer weights, that is implemented through the

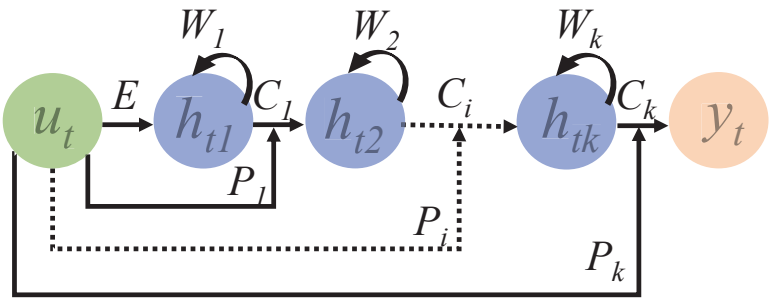

Figure 4.5: The architecture of designed ICRNN with "passthrough" layers.

"Keras" package in Python.

In summary, ICRNN is able to make full use of the powerful representation capability of neural network by adopting the architecture of RNN with "passthrough" layers. Meanwhile, the convexity of RNN is also achieved through the non-negative layer weighting, which could guarantee the solution of control optimization is unique and globally optimal. Therefore, ICRNN offers us a novel idea to deal with optimal control problems in complex systems through trading off the modeling accuracy and control tractability.

Remark: In this work the application object is a single-rack DC with rack-based cooling architecture, and only a small number of variables are used to represent the thermal system. It seems that the multi-input-single-output thermal system is easy to model through the ICRNN. However, the development of the ICRNN is still difficult owing to the structure design, hyper-parameter tuning and following combination of EWC algorithm. Furthermore, this ICRNN could also be extended to the thermal modeling of the whole DC efficiently after the validation.

4.4.2 Algorithm of elastic weight consolidation

In industrial systems, it is quite difficult to collect the data of all working conditions to train a model with high accuracy. Therefore, sometimes we have to unpredictably train the neural networks through continual learning or incremental learning. However, during the process of learning how to deal with new task, the neural networks always forget the capability of handling previous task [42]. The problem in computer science is called catastrophic forgetting, which greatly degrades the performance of neural networks in continual learning [43, 44, 45].

In order to address this problem, an algorithm of elastic weight consolidation has been proposed recently. This algorithm is inspired through the synaptic consolidation in human brain. The synaptic consolidation could help human to avoid forgetting the ability of dealing with previously learned tasks through decreasing the plasticity of synapses [30]. Similarly, to overcome the catastrophic forgetting in continual learning, we develop an approach to restrict the important layer parameters of neural networks to the parameter region of previous task and reduce the the plasticity of such parameters. The detailed methods of implementing the constraints and figuring out the important parameters are introduced in the following section.

Generally, the training of a neural network is the procedure of optimizing layer parameters and adapting the model to specific tasks. As many configurations of layer parameters would lead to the same performance, we can figure out a unique parameter region for each task. For the similar tasks (such as the classifications of different animals, or the regression of data from different working conditions but in the same system), there is probably a overlap between the parameter regions. The diagram of different parameter regions are shown in Figure 4.6. So when we retrain a

neural network of task A based on the data of task B without any penalty, the weights θ_A^* will change through the green arrow in Figure 4.6 to the parameter region for task B completely and the model would forget the solution of task A. To overcome the catastrophic forgetting, now we would like to figure out an elastic training method to guide the weights θ_A^* toward the overlap region along the red arrow. Thus, the retained neural network is capable of dealing with task B and protesting the model performance in task A simultaneously. Furthermore, it is unnecessary to impose the constraints on all the parameters, otherwise the weights θ_A^* will change along the blue arrow and the model fails to learn task B.

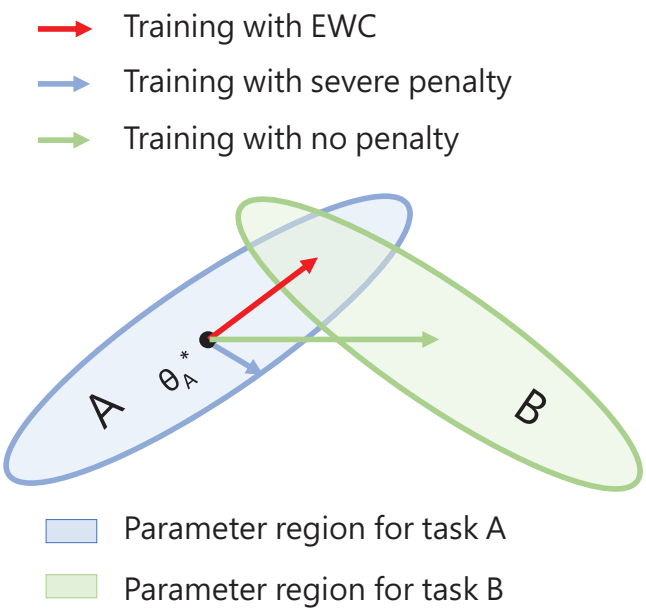

Figure 4.6: The detailed diagram of EWC algorithm.

In order to make clear the implementation of constraints and justification of weights importance for previous task, it is better to investigate the model training from the perspective of probability. Then the optimization of neural network parameters is converted to the calculation of their most probable values based on

the training data set. The detailed deduction based on Bayes' rule is provided as following:

$$L(\theta) = p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{p(D)}, \tag{4.4.5}$$

where θ means the total parameters, D is the data set, $L(\theta)$ is defined as the loss function, $p(\theta \mid D)$ stands for the conditional probability of the layer parameters, $p(D \mid \theta)$ denotes the conditional probability of data set, $p(\theta)$ is the prior probability of parameters, and $p(\theta)$ represents the prior probability of data set.

Now we divide the data set D into two independent parts: D_A (task A) and D_B (task B), then the equation is transformed to:

$$\begin{aligned} p(\theta \mid D) = p(\theta \mid D_A, D_B) &= \frac{p(D_A, D_B \mid \theta)p(\theta)}{p(D_A, D_B)} \\ &= \frac{p(D_A \mid \theta)p(D_B \mid \theta)p(\theta)}{p(D_A)p(D_B)} = \frac{p(\theta \mid D_A)p(D_B \mid \theta)}{p(D_B)}. \end{aligned} \tag{4.4.6}$$

After taking the logarithm of both side, the equation is expressed as below:

$$\log p(\theta \mid D) = \log p(\theta \mid D_A) + \log p(D_B \mid \theta) - \log p(D_B). \tag{4.4.7}$$

In this equation, the posterior probability of the layer parameters based on the whole data set $p(\theta \mid D)$ is still regarded as the final loss function $L(\theta)$, $p(D_B \mid \theta)$ is considered as the loss function for task B $L_B(\theta)$, $p(\theta \mid D_A)$ is treated as the important information of task A, and $p(D_B)$ is a constant. Then the optimization of

layer parameters θ could be represented as:

$$\begin{aligned}\theta &= \arg\ \max_{\theta} L(\theta) = \arg\ \max_{\theta} \log p(\theta \mid D) \\ &= \arg\ \max_{\theta}(-L_B(\theta) + \log p(\theta \mid D_A)) \\ &= \arg\ \min_{\theta}(L_B(\theta) - \log p(\theta \mid D_A)),\end{aligned} \tag{4.4.8}$$

where the value of probability $p(D_B \mid \theta)$ is always less than one, so the negative of loss function for task B is used to substitute $\log p(D_B \mid \theta)$.

The posterior probability of parameters for task A $p(\theta \mid D_A)$ is essential to implement EWC, as it contains many important memories of task A. Additionally, the posterior probability $p(\theta \mid D_A)$ is quite difficult to compute directly, so here we assume the probability as a function with respect to θ and implement second-order Taylor expansion at $\theta = \theta_A^*$. θ_A^* represents the optimal layer parameter for task A and it has been acquired in the previous training.

$$f(\theta) = \log p(\theta \mid D_A), \tag{4.4.9}$$

$$f(\theta) = f(\theta_A^*) + \frac{f'(\theta_A^*)}{1!}(\theta - \theta_A^*) + \frac{f''(\theta_A^*)}{2!}(\theta - \theta_A^*)^2 + \epsilon(\theta_A^*), \tag{4.4.10}$$

where $\epsilon(\theta_A^*)$ stands for the high-order term in Taylor expansion and it would be neglected in the following calculation. Besides, the first-order derivative $f'(\theta_A^*)$ should be zero, as the optimal parameter θ_A^* is the extreme value of the function. At last the equation could be simplified as below:

$$f(\theta) \approx f(\theta_A^*) + \frac{f''(\theta_A^*)}{2}(\theta - \theta_A^*)^2. \tag{4.4.11}$$

Then the optimization of parameter θ is transformed as:

$$\theta = \arg\ \min_{\theta}(L_B(\theta) - \frac{f''(\theta_A^*)}{2}(\theta - \theta_A^*)^2). \tag{4.4.12}$$

Since the layer parameter θ in neural network is a matrix, the second-order derivative of the function $f''(\theta_A^*)$ should be a matrix as well and it is named Hessian matrix in matrix theory. Generally the computation of Hessian matrix is very complex for neural networks, which would significantly increase the training time. Therefore, the diagonal of the Fisher information matrix is adopted in this work to approximate the Hessian matrix [46, 47]. Finally, the calculation of parameter θ in EWC is illustrated as:

$$\theta = \arg\ \min_{\theta}(L_B(\theta) + \sum_i \frac{\kappa}{2} F_i(\theta_i - \theta_{A,i}^*)^2), \tag{4.4.13}$$

where κ is the coefficient, and F_i is the Fisher information matrix which denotes how important the parameters for the previous task A. It noteworthy that the Fisher information matrix is the negative expectation of Hessian matrix, thus the minus in Eq. (4.4.12) changes to plus sign in Eq. (4.4.13).

For the learning of multiple tasks, for example the learning of three tasks (task A, task B and task C), the algorithm of EWC would slow down the learning of the important parameters for task A and task B. Then the loss function could be expressed as following:

$$L(\theta) = L_C(\theta) + \frac{1}{2}\sum_i [\kappa_A F_{A,i}(\theta_i - \theta_{A,i}^*)^2 + \kappa_B F_{B,i}(\theta_i - \theta_{B,i}^*)^2]. \tag{4.4.14}$$

4.5 Experimental-data-based Simulation results

All the training and testing data is collected in a real single-rack DC with rack-based cooling architecture, and the experimental devices are shown in Figure 4.7. The experimental system includes nearly thirty servers, a RMCU, workload monitor, several air temperature sensors, water temperature sensor, water flow meter, power source, some pipes and cables. The micro controller board in RMCU comprises a Raspberry Pi Zero, Arduino board (Nano), voltage converter, and relays. The Raspberry Pi Zero is used as the master micro-controller to implement the main control algorithm. The Arduino board is applied as the slave micro-controller to collect data from sensors and send the control commands to actuators. The design of slave micro-controller is to reduce the computational load of Raspberry Pi. Besides, the relays are dedicated to the control of electrical motorized ball valve.

In this section, we present two simulation cases. The first case is implemented to verify the potential performance of proposed method against catastrophic forgetting in sequence learning. The ICNN would be trained based on the data collected from a specific working condition (we name the dataset data-A). Then the ICNN is retrained through the traditional method (SGD) and the method with EWC based on the data collected from another working condition (we name the second dataset data-B). Finally, the ICNN retrained by traditional method and ICNN retrained via the method with EWC is compared based on the data-A. The simulation is conducted to reveal whether the algorithm of EWC could allow the ICNN to learn features from the data in different working conditions without catastrophic forgetting.

The second case is carried out to validate the proposed ICNN trained through

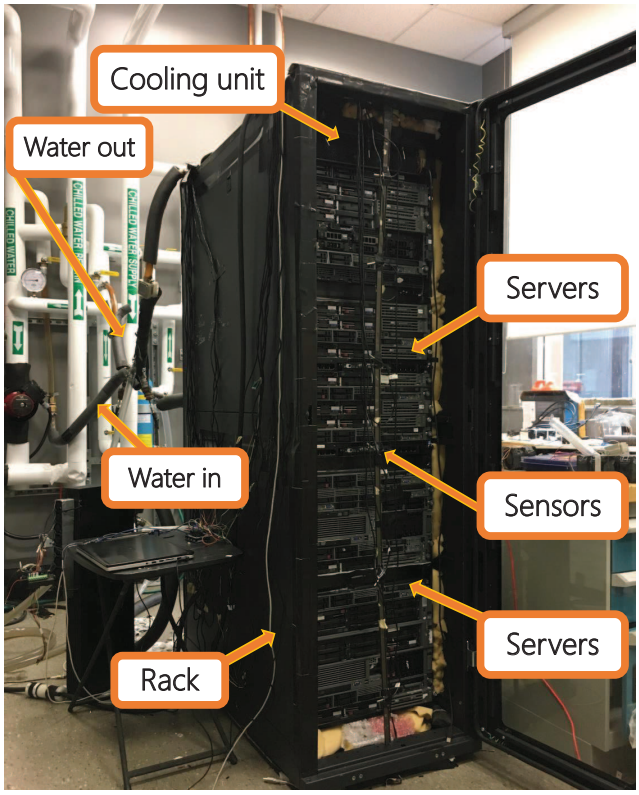

Figure 4.7: Experimental DC with rack-based cooling architecture.

EWC is robust to the noisy training data during sequence learning. Since the algorithm of EWC is developed to avoid the catastrophic forgetting in continual learning, it should remember the important features of previous clean data and be resistant to the noises in following training data. Therefore, the simulation concerning the robustness of our developed method is conducted to explore whether the EWC algorithm is efficient to eliminate the negative effects of noisy data during the retraining process.

4.5.1 Case I: Simulation for validating the ability of overcoming catastrophic forgetting

In this case, the ICNN is trained through data-A first and then retrained based on data-B by two different methods: the normal training and EWC-based training. The basic inputs information of data-A and data-B is represented in Figure 4.8 and Figure 4.9 respectively. Additionally, 80% data in each dataset is use for training and the rest of data is used for validation. The first working condition is used to simulate the common situation in DCs, where the IT workload remains constant, the air flow rate and water flow rate change randomly. The constant IT workload means the network traffic is stable, and it is quite normal for most of the time. While, the varying air flow rate and water flow rate indicate the controller of RMCU works for regulating the temperature in the rack.

The second working condition is selected to simulate the extreme situation in DCs, in which the IT workload, air flow rate and water flow rate change simultaneously. In DCs, most of the racks would run at full power generally, and a small part of racks stay in standby mode. Only when the network traffic is congested, the idle racks would start to work and thus result in varying the IT workload. This type of working condition is irregular, so the data is difficult to collect and thus we need to train the neural networks in sequence.

The comparison of temperature simulation is shown in Figure 4.10. As is seen in this figure, the red curve denotes the true value of temperature in the rack, and the green and black curves represent the performances on the testing set of data-A. The green curve means the performance of model trained through EWC algorithm overcoming catastrophic forgetting, and the black curve stands for the performance

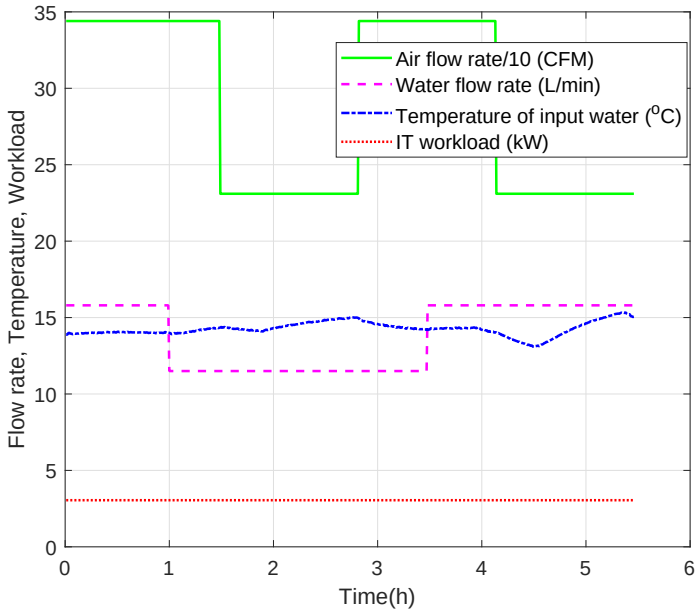

Figure 4.8: Basic input information of the data-A collected in the first working condition.

of model trained by normal method.

In this figure, the prediction of model trained by EWC algorithm could follow the true value at beginning, nevertheless, the prediction could also precisely track the true value even on the inflection point. However, the prediction of model trained through normal method fails to follow the reference in most of the time. Although the prediction of model trained through normal method is more accurate in a certain period, the prediction accuracy cannot remain stable. The simulation result in this case demonstrates that the catastrophic forgetting in continual learning does degrade the model performance with respect to the previous tasks, while the training method with EWC is able to overcome catastrophic forgetting successfully and keep the prediction accuracy for both previous task and current task.

In addition, the prediction errors of model trained by EWC algorithm and model

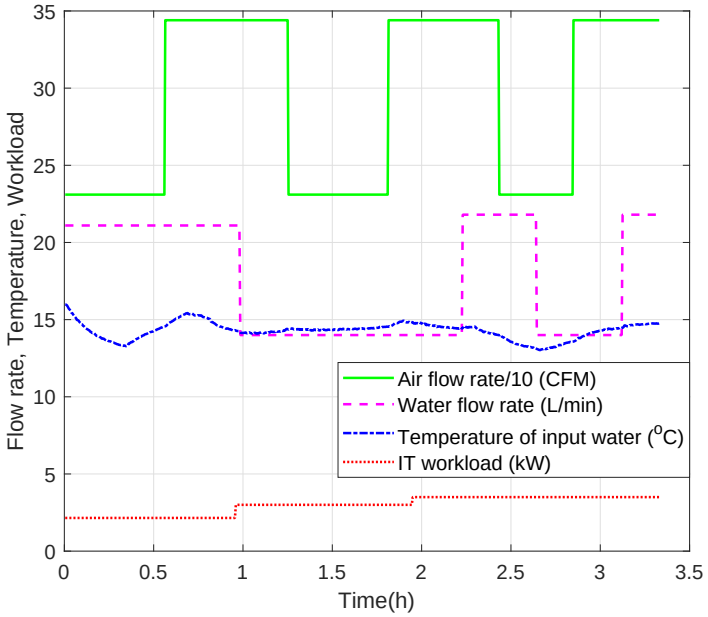

Figure 4.9: Basic input information of the data-B collected in the second working condition.

trained through normal method are presented in Figure 4.11. As we can see in this figure, the prediction error of model trained by EWC algorithm is generally within the range of 0.1 degree to -0.1 degree. The prediction errors illustrate that the proposed ICNN achieves significant improvement in thermal modeling in DCs, and further prove the ICNN is accurate enough to apply for future controller design.

4.5.2 Case II: Simulation for validating the ability of noise immunity

The second case in conducted to test the performance of ICNN model trained through EWC algorithm under noisy training data. In this case, the validation data is collected from the normal working condition which is similar to the data-A in the first case.

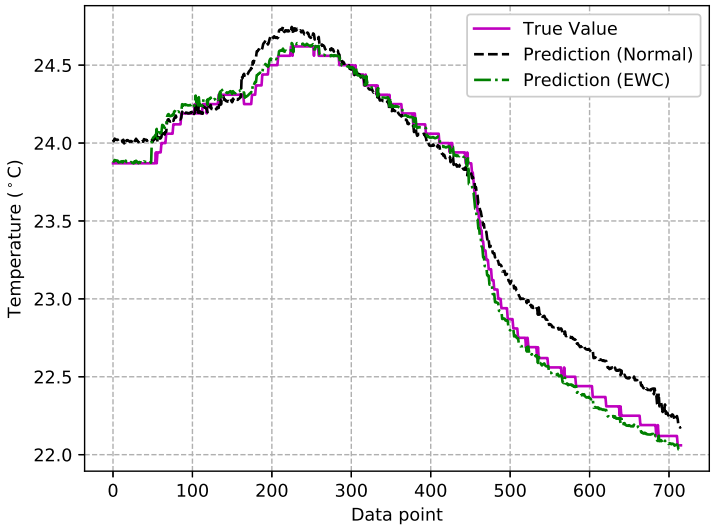

Figure 4.10: The comparison of temperature simulation under different training methods in case I.

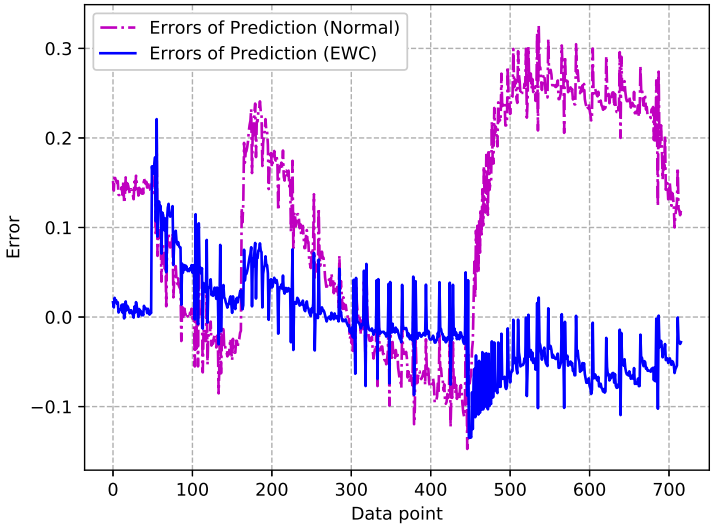

Figure 4.11: The comparison of predicted temperature errors under different training methods in case I.

At beginning, 40% of validation data is used for the first-stage training. Then 40% of validation data would be added the Gaussian white noise and utilized for the second-stage retraining. The retraining process is implemented through the normal method and EWC algorithm like the first case. Finally the rest of validation data is employed to test the ICNN model based on normal method and method with EWC algorithm respectively. Besides, the input information of validation data is presented in Figure 4.12, and the synthetic noisy training data is described in Figure 4.13.

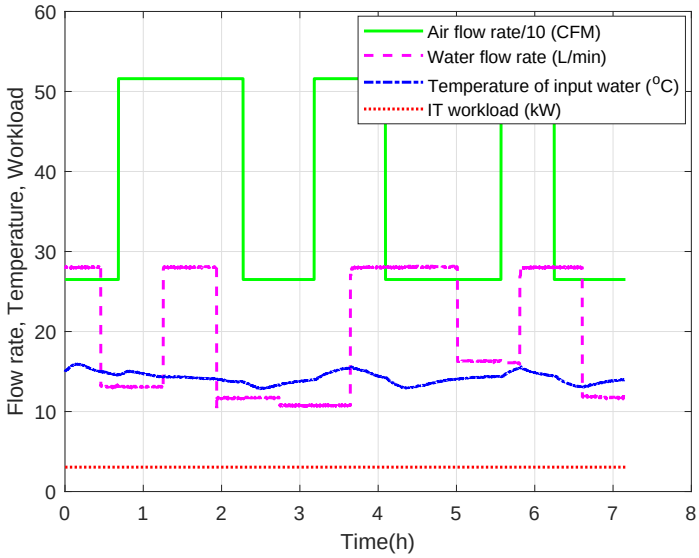

Figure 4.12: Basic input information of validation data in case II.

The comparison of simulation results for the second case is depicted in Figure 4.14. In this figure, the performance of model trained through EWC algorithm is still better than the model trained by normal method. Particularly, the prediction of EWC-based model could track the true value with small error immediately, while the prediction of model trained through normal method cannot follow the slowly varying true value for the first half. The poor simulation result of model trained through

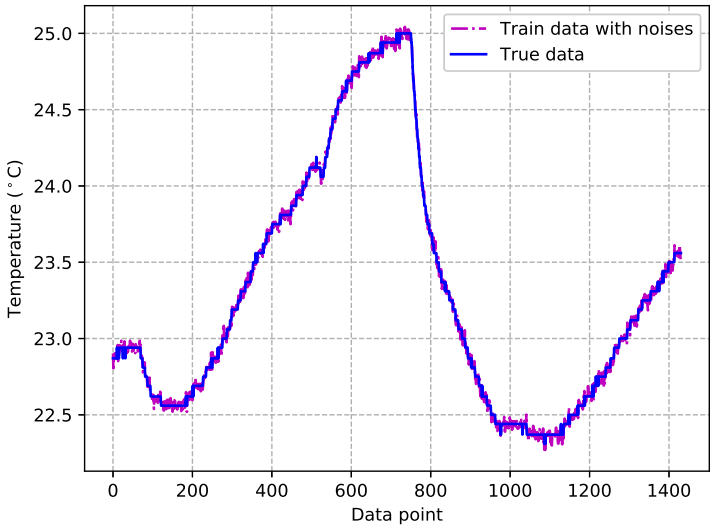

Figure 4.13: Synthetic noisy data for second-stage retraining.

normal method illustrates that the noisy data has significant negative effect on the prediction performance in continual learning. Nonetheless, the accurate prediction of EWC-based model demonstrates that the EWC algorithm could remember the important features of clean data in the first-stage training, and thus relatively prevent the disturbance of noise in the second-stage retraining data.

Then the The comparison of predicted temperature errors is shown in Figure 4.15. As is shown in this figure, the overall predicted error of model trained through EWC algorithm is smaller than the model trained via normal method. The comparison of predicted errors reveals the superiority of ICNN trained by EWC algorithm with new clarity. Moreover, the predicted error of model trained through EWC is still within 0.1 degree, which indicates it is effective to use our proposed ICNN model for following control design.

The validation of two cases represent that the algorithm of EWC allows features

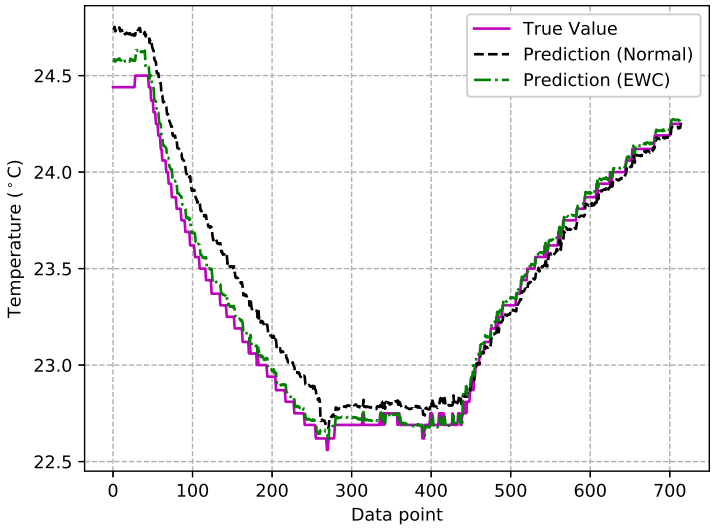

Figure 4.14: The comparative temperature results under different training methods in case II.

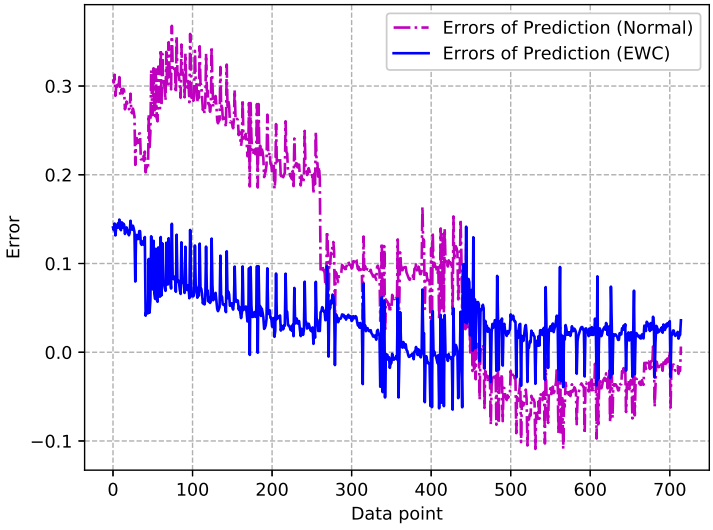

Figure 4.15: The comparison of predicted temperature errors under different training methods in case II.

of previous training data to be protected during the relearning with newly collected data, and thus avoid the catastrophic forgetting in continual learning. Besides, the second validation also explore the robustness of ICNN trained by EWC algorithm when the latest collected training data is noisy, and prove that our proposed method is efficient to resist the negative effect of noisy data during the continual learning. Finally, this method also provide us a novel idea to deal with the noisy labels in deep learning.

4.6 Conclusion

This paper deals with the thermal modeling issue for DCs with rack-based cooling architecture. In order to accurately describe the thermal dynamics in this system, an ICNN model is proposed in this work. The ICNN is designed to bridge the gap between control tractability and representation capability of neural networks, and thus could be applied for further optimal control. Moreover, the problem of catastrophic forgetting in continual learning is also considered in this paper. Inspired by the synaptic consolidation in neurobiology, an algorithm of EWC is developed to elastically restrict the important layer weights for previous task, thereby overcoming the catastrophic forgetting in following retraining. The results of simulation illustrate the effectiveness of our designed method with respect to the avoidance of catastrophic forgetting. Nevertheless, the results also explore the robustness of our method against noisy training data, and achieve excellent performance. In the future, an effective controller would be developed based on this ICNN model to implement the thermal management in DCs.

Bibliography

[1] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renew. Sustain Energy Rev.*, vol. 31, pp. 622–638, Mar. 2014.

[2] T. Lu, X. Lü, M. Remes, and M. Viljanen, "Investigation of air management and energy performance in a data center in finland: Case study," *Energy and Buildings*, vol. 43, no. 12, pp. 3360–3372, Dec. 2011.

[3] H. Zhang, S. Shao, H. Xu, H. Zou, and C. Tian, "Free cooling of data centers: A review," *Renew. Sustain Energy Rev.*, vol. 35, pp. 171–182, Jul. 2014.

[4] N. Hassan, M. M. K. Khan, and M. Rasul, "Temperature monitoring and cfd analysis of data centre," *Procedia Engineering*, vol. 56, pp. 551–559, 2013.

[5] S. Asgari, S. MirhoseiniNejad, H. Moazamigoodarzi, R. Gupta, R. Zheng, and I. K. Puri, "A gray-box model for real-time transient temperature predictions in data centers," *Applied Thermal Engineering*, p. 116319, Nov.

[6] S. Asgari, H. Moazamigoodarzi, P. J. Tsai, S. Pal, R. Zheng, G. Badawy, and I. K. Puri, "Hybrid surrogate model for online temperature and pressure predictions

in data centers," *Future Generation Computer Systems*, vol. 114, pp. 531–547, Jan.

[7] M. Zapater, J. L. Risco-Martín, P. Arroba, J. L. Ayala, J. M. Moya, and R. Hermida, "Runtime data center temperature prediction using grammatical evolution techniques," *Appl. Soft Comput.*, vol. 49, pp. 94–107, 2016.

[8] L. Parolini, B. Sinopoli, B. H. Krogh, and Z. Wang, "A cyber–physical systems approach to data center modeling and control for energy efficiency," *Proc. IEEE*, vol. 100, no. 1, pp. 254–268, Jan. 2012.

[9] L. Parolini, B. Sinopoli, and B. H. Krogh, "Model predictive control of data centers in the smart grid scenario," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 10 505–10 510, Jan. 2011.

[10] B. Durand-Estebe, C. Le Bot, J. N. Mancos, and E. Arquis, "Data center optimization using PID regulation in CFD simulations," *Energy and Buildings*, vol. 66, pp. 154–164, Nov. 2013.

[11] J. Deng, L. Yang, X. Cheng, and W. Liu, "Self-tuning PID-type fuzzy adaptive control for CRAC in datacenters," in *International Conference on Computer and Computing Technologies in Agriculture*, pp. 215–225, 2013.

[12] H. Cheung, S. Wang, C. Zhuang, and J. Gu, "A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation," *Appl. energy*, vol. 222, pp. 329–342, Jul. 2018.

[13] H. Moazamigoodarzi, P. J. Tsai, S. Pal, S. Ghosh, and I. K. Puri, "Influence of cooling architecture on data center power consumption," *Energy*, vol. 183, pp. 525–535, Sep. 2019.

[14] R. Zhou, Z. Wang, C. E. Bash, and A. McReynolds, "Modeling and control for cooling management of data centers with hot aisle containment," in *Proceedings of ASME 2011 International Mechanical Engineering Congress & Exposition*, 2011.

[15] M. Ogawa, H. Fukuda, H. Kodama, H. Endo, T. Sugimoto, T. Kasajima, and M. Kondo, "Development of a cooling control system for data centers utilizing indirect fresh air based on model predictive control," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 132–137, 2015.

[16] M. Jonas, R. R. Gilbert, J. Ferguson, G. Varsamopoulos, and S. K. Gupta, "A transient model for data center thermal prediction," in *2012 International Green Computing Conference (IGCC)*, pp. 1–10. IEEE, 2012.

[17] V. A. Tsachouridis and T. Scherer, "Data centre adaptive numerical temperature models," *Trans. Inst. Meas. Control*, vol. 40, no. 6, pp. 1911–1926, Mar. 2018.

[18] W. A. Abdelmaksoud, H. E. Khalifa, T. Q. Dang, R. R. Schmidt, and M. Iyengar, "Improved CFD modeling of a small data center test cell," in *12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 1–9. IEEE, 2010.

[19] H. Moazamigoodarzi, S. Pal, S. Ghosh, and I. K. Puri, "Real-time temperature predictions in IT server enclosures," *Int. J. Heat Mass Transf.*, vol. 127, pp. 890–900, Dec. 2018.

[20] J. Athavale, M. Yoda, and Y. Joshi, "Comparison of data driven modeling approaches for temperature prediction in data centers," *Int. J. Heat Mass Transf.*, vol. 135, pp. 1039–1052, Jun. 2019.

[21] E. Samadiani, Y. Joshi, H. Hamann, M. K. Iyengar, S. Kamalsy, and J. Lacey, "Reduced order thermal modeling of data centers via distributed sensor data," *ASME J. Heat Transfer*, vol. 134, no. 4, Feb. 2012.

[22] R. Ghosh and Y. Joshi, "Error estimation in POD-based dynamic reduced-order thermal modeling of data centers," *Int. J. Heat Mass Transf.*, vol. 57, no. 2, pp. 698–707, Feb. 2013.

[23] K. Jiang, S. Shi, H. Moazanigoodarzi, C. Hu, S. Pal, and F. Yan, "Temperature distribution estimation via data-driven model and adaptive kalman filter in modular data centers," *Proc. IMechE, Part I: J. Systems and Control Engineering*, p. 0959651820903201, Apr. 2020.

[24] Z. Song, B. T. Murray, and B. Sammakia, "A dynamic compact thermal model for data center analysis and control using the zonal method and artificial neural networks," *Appl. Therm. Eng.*, vol. 62, no. 1, pp. 48–57, Jan. 2014.

[25] Z. Song, B. T. Murray, and B. Sammakia, "Airflow and temperature distribution optimization in data centers using artificial neural networks," *Int. J. Heat Mass Transf.*, vol. 64, pp. 80–90, Sep. 2013.

[26] S. Tashiro, Y. Nakamura, K. Matsuda, and M. Matsuoka, "Application of convolutional neural network to prediction of temperature distribution in data centers," in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, pp. 656–661. IEEE, 2016.

[27] E. Chung, J. Fowers, K. Ovtcharov, M. Papamichael, A. Caulfield, T. Massengill, M. Liu, D. Lo, S. Alkalay, M. Haselman *et al.*, "Serving DNNs in real time at datacenter scale with project brainwave," *IEEE Micro*, vol. 38, no. 2, pp. 8–20, Mar. 2018.

[28] Y. Chen, Y. Shi, and B. Zhang, "Optimal control via neural networks: A convex approach," in *International Conference on Learning Representations*, 2019.

[29] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *International Conference on Machine Learning*, pp. 146–155, 2017.

[30] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[31] T. Ding, Z. guang He, T. Hao, and Z. Li, "Application of separated heat pipe system in data center cooling," *Appl. Therm. Eng.*, vol. 109, pp. 207–216, Oct. 2016.

[32] S. MirhoseiniNejad, H. Moazamigoodarzi, G. Badawy, and D. G. Down, "Joint data center cooling and workload management: A thermal-aware approach," *Future Gener. Comput. Syst.*, vol. 104, pp. 174–186, Mar.

[33] H. Moazamigoodarzi, R. Gupta, S. Pal, P. J. Tsai, S. Ghosh, and I. K. Puri, "Modeling temperature distribution and power consumption in it server enclosures with row-based cooling architectures," *Appl. Energy*, vol. 261, p. 114355, Mar.

[34] H. Moazamigoodarzi, S. Pal, D. Down, M. Esmalifalak, and I. K. Puri, "Performance of a rack mountable cooling unit in an it server enclosure," *Therm. Sci. Eng. Prog.*, vol. 17, p. 100395, Jun.

[35] R. Gupta, S. Asgari, H. Moazamigoodarzi, S. Pal, and I. K. Puri, "Cooling architecture selection for air-cooled data centers by minimizing exergy destruction," *Energy*, p. 117625, Jun. 2020.

[36] N. Hirose, R. Tajima, and K. Sukigara, "Mpc policy learning using DNN for human following control without collision," *Adv. Robotics*, vol. 32, no. 3, pp. 148–159, 2018.

[37] Z. Huang, J. Peng, H. Lian, J. Guo, and W. Qiu, "Deep recurrent model for server load and performance prediction in data center," *Complexity*, vol. 2017, 2017.

[38] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters," *Procedia Comput. Sci.*, vol. 125, pp. 676–682, 2018.

[39] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[41] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[42] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Advances in neural information processing systems*, vol. 30, pp. 4652–4662, 2017.

[43] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3400–3409, 2017.

[44] H. Ritter, A. Botev, and D. Barber, "Online structured laplace approximations for overcoming catastrophic forgetting," in *Advances in Neural Information Processing Systems*, pp. 3738–3748, 2018.

[45] K. Lee, K. Lee, J. Shin, and H. Lee, "Overcoming catastrophic forgetting with unlabeled data in the wild," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 312–321, 2019.

[46] R. Pascanu and Y. Bengio, "Revisiting natural gradient for deep networks," in *International Conference on Learning Representations*, 2013.

[47] E. Eskin, A. J. Smola, and S. Vishwanathan, "Laplace propagation," in *Advances in Neural Information Processing Systems*, pp. 441–448. MIT Press, 2004.

Chapter 5

Thermal Modeling in Data Centers through Input Convex Neural Networks with Noisy Data

This chapter is reproduced from *"Thermal Modeling in Data Centers through Input Convex Neural Networks with Noisy Data"*, ***Kai Jiang***, *and Fengjun Yan, submitted to Neurocomputing, 2020.* The author of this thesis is the first author and the main contributor of this publication.

5.1 Abstract

This paper proposed a thermal modeling method for data centers (DCs) with rack-based cooling architecture via a novel input convex neural network (ICNN). This type of deep neural network is convex from inputs to outputs by constraining the layer parameters, thus it could be effectively used for convex optimization in optimal

control issue. Besides, to enhance the robustness with noisy data and avoid overfitting in the training process, an example reweighting algorithm is adopted. In the proposed algorithm, the layer parameters in ICNN are computed with the mini-batch in noisy training data, and then reweighted based on a small portion of the clean dataset. The proposed algorithm is easily performed without any additional hyperparameter tuning, and could be applied for any type of deep neural networks. Finally, the validations demonstrate its effectiveness through experimental data with synthetic noisy data.

Key words: Thermal modeling, Neural Networks, Control modeling, Noisy data, Data centers.

5.2 Introduction

With the ever-increasing demands for computing, networking and communication technologies, a large number of DCs are required to support the services [1, 2, 3]. At the same time, the problem of huge amounts of power consumption in DCs has become a big challenge for us [4]. The studies revealed that US DCs consumed nearly 70 billion kW· h of electricity in 2014, which accounts for two percent of the total power consumption in the US [5]. More seriously, the estimated power consumption of DCs would double in the next five years [6]. Therefore, efficient methods of energy management are urgent needs for current electronic equipments in DCs. Nevertheless, the energy consumption of cooling system in DCs cannot be ignored as well [7]. Cooling system is essential for guaranteeing safe temperature in the facilities, and it will consume one third of the total power consumption in DCs [8, 9]. Thus, productive thermal management techniques based on cooling system are

also necessary to cooperate with energy management in DCs.

No matter for thermal management or energy management in DCs, an accurate model of thermal dynamics is indispensable [10, 11]. In the thermal management, we need a model to forecast the future thermal dynamics and then make proper actions in the cooling system to keep the temperatures at steady state [12, 13, 14, 15]. For energy management system, the thermal model is also required to predict the temperature distribution. Based on such predictions of temperature distribution, appropriate workloads are scheduled for the servers at different location to optimize the energy utilization. Besides, the latest researches about joint thermal and energy management are also heavily dependent upon the accurate model of thermal dynamics in DCs.

In order to build an accurate and efficient temperature model, many approaches have been developed by researchers. The first type of model is physics model, which is built based upon thermodynamics and conservation of matter and energy. In [16], the authors proposed a simplified physics model to represent the temperature distribution in DCs. The advantages of this method include the capabilities of high-speed computation and transient temperature prediction. The researchers in [17] developed a computational fluid dynamics (CFD) model to describe the thermal behaviors in small DCs. The novelty in this paper was the introduction of thermal boundary condition, which could reduce the forecast error significantly. A state-space model was constructed for DC temperature control in [18]. This developed temperature model is adaptive to the input perturbations, and thus more suitable to the complex system. Furthermore, a zonal model for transient temperature prediction was proposed in [19]. Specially, this model was built for a DC equipped with novel rack-based cooling

architecture. This type of cooling system is more efficient than traditional computer room air conditioning (CRAC) system. Although the physics models in DCs are able to represent the thermal system roughly and easier to apply for control design, the modeling processes are really difficult and time-consuming. First, the parameters in fixed thermodynamics formulas are hard to calibrate due to the environmental disturbances, and complicated air flow and heat transfer in the facilities. Secondly, too many assumptions are proposed during the modeling process, which would seriously degrade the model accuracy and even deteriorate the control performance.

Considering the drawbacks of physics models, data-driven models are put forward to accomplish the thermal modeling for DCs [20]. The technique of proper orthogonal decomposition (POD) is a common method used for data-driven thermal modeling in DCs [21, 22]. In these papers, the authors collected real experimental data to calibrate the POD-based data-driven model, and mainly analyzed the errors between experimental data and model predictions. Along the idea of POD, the authors in [23] proposed the approach of partial least square (PLS) to describe the thermal dynamics in DCs. Moreover, the algorithm of fuzzy c-means was also adopted to develop multiple data-driven models to represent the nonlinear system. Compared with the statistical modeling methods, neural networks have higher accuracy for thermal modeling due to their powerful capability of representing the nonlinear systems. Artificial neural networks (ANN) were utilized to forecast the airflow and temperature distribution in DCs [24, 25]. The excellent simulation results demonstrated the superiority of neural networks. Similarly, convolutional neural networks (CNN) were also employed to predict the thermal behaviors in [26]. The proposed method could adaptively learn the local structure, and thus described the underlying system features in DCs. These

investigations revealed that neural networks have good representation capability for thermal modeling in DCs, but most of them are related to temperature prediction and few works focus on using neural networks to deal with control issues in DCs.

According to the literature review, data-driven models based on conventional s-tatistical framework can be easily combined with control design. However, most of them are still in the form of linear models which cannot afford to represent the complex thermal dynamics in large-scale systems such as the DCs. On the contrary, Deep neural networks (DNNs) hold the remarkable representation capability for thermal modeling, while these models commonly have the characteristics of nonconvexity and nonlinearity. Such characteristics usually result in great problems for control design. In order to tackle this problem, the model of ICNN is adopted to make a trade-off between the model accuracy and control tractability. The ICNN model is built under the structure of neural networks and it is convex from inputs to outputs. By applying such method, we could acquire accurate enough thermal models for DCs and solvable solutions of optimization problems in control issues simultaneously.

Although DNNs have powerful representation capability, they also suffer from the problem of overfitting when the training data is noisy. Particularly, in the real industrial experiments, it is quite difficult to collect the clean data owing to the limitations of physics sensors and environmental disturbances. Therefore, in this work, we would like to employ an algorithm to improve the robustness of ICNN and further avoid the overfitting under noisy training data. Inspired by the work in [27], a specific example reweighting algorithm is adopted in our paper to deal with the training set biases. In this method, a small number of clean validation data is required to add in the training set. In each training iteration, the parameters of each layer are

optimized based on the normal mini-batch in training data first, then the important layer parameters would be recalculated through the clean validation data to eliminate the negative effects of noises in the training data. Based on such approach, we could obtain an accurate and robust ICNN with the capabilities of representing complicated thermal systems and counteracting noises in training set simultaneously. Finally, the proposed method is utilized in a single-rack DC with rack-based cooling architecture, and validated through the real experimental data.

The rest of this paper is organized as follows. Section II shows the detailed structure of DC. Section III presents the modeling methodology including the construction of ICNN and example reweighting algorithm. Section IV mainly depicts the simulation results and discussions. At last, the conclusion is shown in Section V.

5.3 Fundamental of DC with rack-based cooling architecture

Cooling system is an essential section of the DCs, which is employed to maintain suitable environmental temperature in the facilities and further guarantee the safe operation of electronic devices [3]. The most commonly used cooling approach in DCs is air cooling due to the reliability and lower cost compared with liquid cooling or novel phase-change-material cooling [28]. For the air cooling architecture, three design concepts including room-based, row-based and rack-based architecture are developed to implement the cooling task. The room-based cooling system is just like the conventional air-condition. The cold air is directly delivered to the whole room for temperature regulation. The row-based cooling system is designed to provide cold

air to a row of racks with lots of servers, and further control the temperature. The last one is rack-based cooling system, which is developed to cool the servers in a single rack. The detailed configurations of three architectures are shown in Figure 5.1.

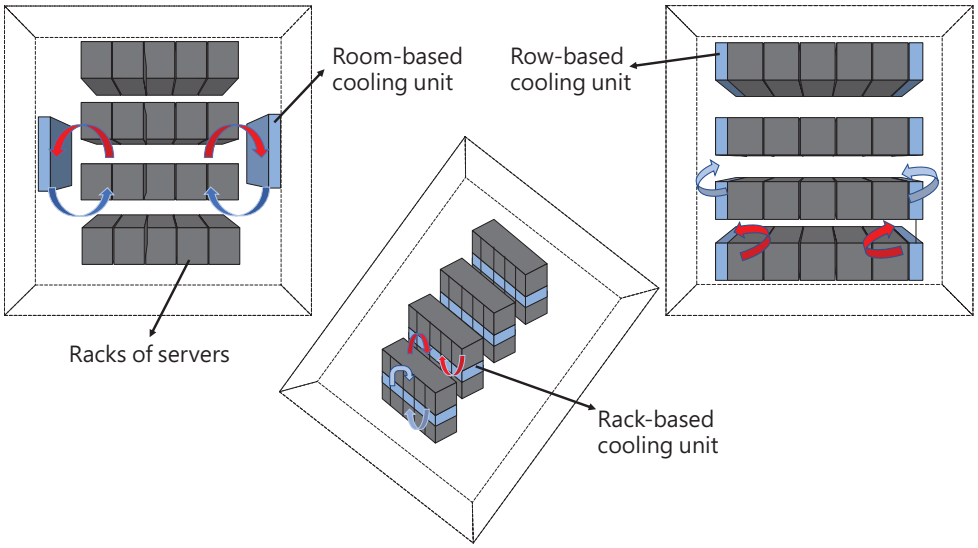

Figure 5.1: Configurations of DCs with room-based, row-based and rack-based cooling architecture

.

The room-based cooling system is widely adopted in many DCs due to the competitive price, convenient installation, and easy operation. However, with the increase in the construction of new DCs, the shortcoming of poor efficiency in room-based cooling system appears gradually. Because of the hot air recirculation and cold air bypass in the facility, the room-based cooling system cannot cool the devices efficiently and thus increase the overall power consumption. Conversely, row-based and rack-based cooling system perform much better on energy efficiency. The shorter airflow paths and distinct hot and cold chambers in DCs with enclosed row-based and rack-based cooling architecture could significantly increase the cooling efficiency and reduce the

power consumption. Moreover, such cooling architectures with short airflow path also allow us to precisely control the cooling load in response to the disturbances of IT servers and cooling unit, and thus keep the temperature at a steady state. For row-based cooling system and rack-based cooling system, the customers could make different choices according to the specific requirements. Besides, based on the latest investigations in our group [29], the exergy destruction of rack-based cooling system is lower. Thus, our studies in this paper would focus on the DC with rack-based cooling architecture. The configuration and supporting facilities (frontal and profile) are presented in Figure 5.2 and Figure 5.3 respectively.

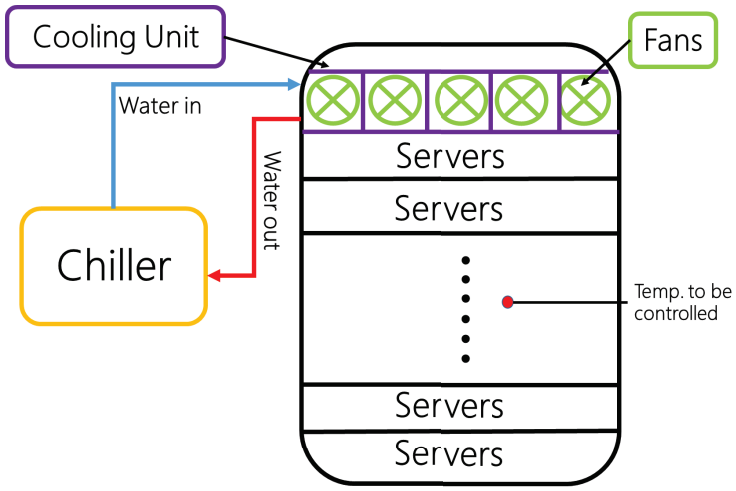

Figure 5.2: Frontal configuration and supporting facilities of DC with rack-based cooling architecture.

The main working principle of rack-based cooling unit is the heat transfer between the hot air in the rack and chilled water from outside chiller. The hot air is extracted from the back of rack to the front via the fans, and the circulating cold water is driven and chilled by the chiller. Then the heat transfer is accomplished in a heat exchanger mounted in the cooling unit. By such method, the heat generated through

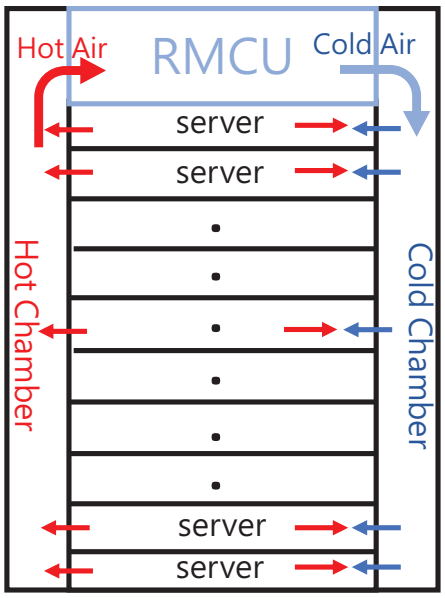

Figure 5.3: Lateral configuration of DC with rack-based cooling architecture.

the servers and other auxiliary devices is taken out by the chiller water. The rack-based cooling unit designed by our group is shown in Figure 5.4, which is named rack mountable cooling unit (RMCU). The RMCU includes several fans, heat exchanger, power source, valve, pipes, control unit and some cables. According to the arranging density of servers, the RMCU could be installed at the top, middle or bottle of the rack.

In the perspective of control theory, the control objective of the system is the temperature in the rack. One of the inputs is the airflow rate, which is determined through the fans; the other input is the chilled water flow rate, which is controlled by the valve. Therefore, the thermal control system in the rack is a multi-input-single-output (MISO) system. Furthermore, the temperature of inlet chilled water and the IT workload of servers are regarded as measured disturbances. The temperature of inlet water is measured by a water temperature meter and it depends on the

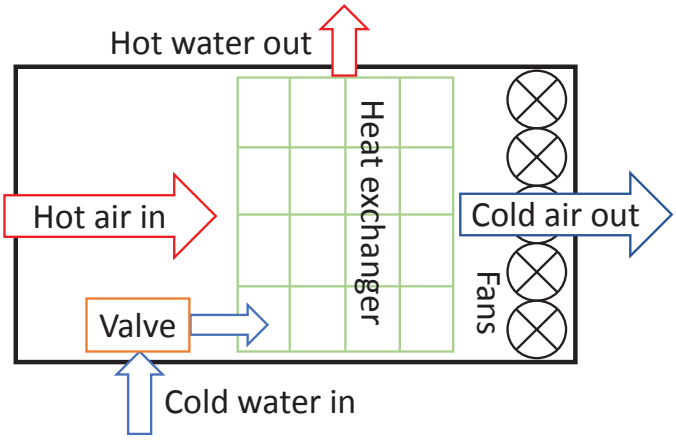

Figure 5.4: Internal structure of rack mountable cooling unit.

outside chiller, which cannot be controlled through the control unit in the RMCU. The discrete-time dynamic model could be represented as below:

$$T_t = f(T_{t-1}, U_t, D_t), \tag{5.3.1}$$

where

$$U_t = [F_{a,t}, F_{w,t}], \tag{5.3.2}$$

$$D_t = [T_{w,t}, \Phi_t]. \tag{5.3.3}$$

Here f is the state function, T represents the temperature in the rack, U denotes the inputs including air flow rate F_a and water flow rate F_w, and D stands for the measured disturbances including input water temperature T_w and IT workload Φ.

5.4 Modeling algorithm of input convex neural networks with robust learning

5.4.1 Input convex neural networks

Neural networks are known as its powerful representation capability of complex systems, while they are usually difficult to be used as dynamic model for control design due to the nonlinearity and nonconvexity. To tackle the control-tractable problem of neural networks, the approach of ICNN is developed. This method doesn't change the internal structure of neural networks too much, but just constrain the parameters of layers in neural networks during the training process. Based on such trick, the neural networks are guaranteed to be convex from inputs to outputs. The detailed introduction is provided as follows.

With the development of deep learning, different types of neural networks are proposed for different applications. In the neural networks, convolutional neural network (CNN) and recurrent neural networks (RNN) are most famous. Generally, CNN is widely used for image processing, segmentation and classification owing to the ability of capturing spatial features, while RNN is usually applied for the tasks of time series forecasting such as natural language processing or system modeling due to the advantage of sequential information acquisition.

As our target in this work is to build a dynamic model to describe the thermal system in DC, RNN is selected as the basic model structure. The RNN is evolved through the artificial neural networks, which consists of input layer, multiple hidden layers and output layer as well. The special feature of RNN is the recurrent connection on the hidden neurons, which means the previous memories of hidden neurons would

be employed for themselves again in the following steps [30]. Such looping constraint guarantees that sequential information would be captured successfully in the training data. The detailed diagram of common RNN is presented in Figure 5.5, and the formulas are given as below:

$$h_t = \sigma_h(Wh_{t-1} + Eu_t + b_h), \tag{5.4.1}$$

$$y_t = \sigma_y(Ch_t + b_y), \tag{5.4.2}$$

where u_t stands for the inputs, h_t means the hidden neurons and y_t is the outputs respectively. $\sigma(\cdot)$ expresses the activation function. The common activation functions used in RNN include "Sigmoid", "Tanh", "ReLU" and some other modified functions. W, C, E and b are defined as the layer weights and biases. Moreover, if the RNN is constructed with multiple hidden layers or some other specific layers, it is called deep RNN.

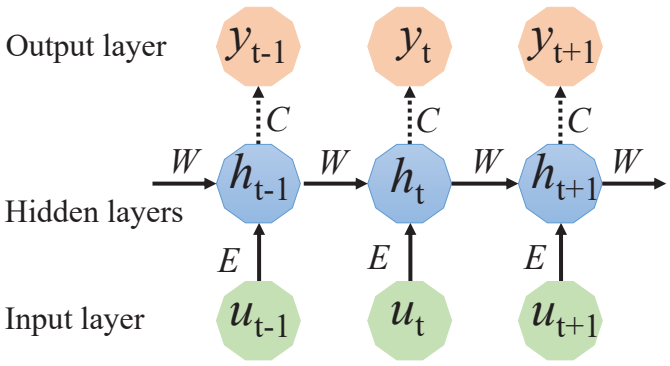

Figure 5.5: The brief structure of traditional RNN.

In order to derive convex RNNs, an effective method of constraining layer weights is proposed. This approach proposes that all the layer weights should be restricted

to be non-negative during the training process, and the activation function should be convex and nondecreasing. These constraints are based on the following proposition:

Proposition 1. The developed neural network is guaranteed to be convex should satisfy that all parameters on hidden layers C_i and W_i are non-negative, and all activation functions σ_i should be convex and nondecreasing (e.g. ReLU).

The proof of this proposition is quite easy, and it has been provided in [31, 32] based on the following facts in [33]:

1) Affine function is still convex when its domain is a convex set.

2) Nonnegative weighted sums of convex functions are still convex.

3) The composition of a convex function and convex nondecreasing function is still convex.

As the structure of neural network in this work is the RNN, we could also name the adopted neural network input convex recurrent neural network (ICRNN). The constraints of layer weights could efficiently ensure that the RNN is convex, however, such constraints on layer weights also degrade the performance of RNN, which betrays our original idea of leveraging the powerful representation capability of RNN. Therefore, the "passthrough" layers are introduced to ICRNN. The "passthrough" layers are proposed to connect the input layer and hidden layers directly to make full use of input information, and thus improve the accuracy of ICRNN [34, 35]. Furthermore, the "passthrough" layers don't effect the convexity properties of ICRNN. The detailed formulas of $k+1$-layer deep ICRNN with "passthrough" layers are shown as following and the architecture of reformed deep ICRNN is presented in Figure 5.6.

$$h_{t,i} = \sigma_i(W_i h_{t-1,i} + C_i h_{t,i-1} + P_i u_t + b_{t,i}), \tag{5.4.3}$$

$$y_t = \sigma_k(C_k h_{t,k} + P_k u_t + b_{t,k}), \tag{5.4.4}$$

where P is defined as the weights of our employed "passthrough" layers.

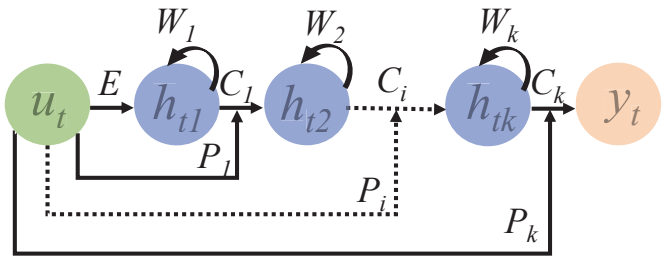

Figure 5.6: The architecture of designed ICRNN with "passthrough" layers.

The training method used for ICRNN is also similar to the general training algorithm. First the approaches of forward propagation and back propagation are utilized to calculate the errors between model predictions and true values. Then the algorithm of stochastic gradient descent (SGD) is adopted to implement the optimizations. The key point here is that the optimized weights should be restricted to be non-negative. Additionally, the approximate capability and computational load of ICRNN compared with multiple affine functions have been provided in [32]. In this work, we wouldn't discuss this issue.

Remark: The application in this paper is a single-rack DC with rack-based cooling architecture, and only a small amount of variables are utilized to describe the thermal behavoirs. Ostensibly, it seems that the thermal system of single-rack DC is easy to represent through the neural network. However, the development of the ICRNN is still complicated due to the model structure design, hyper-parameter tuning and following combination with reweighting example algorithm. Additionally, the ICRNN could easily be extended to the thermal modeling of the whole DC in the

future.

5.4.2 Reweighting example algorithm for robust deep learning

Both of the noisy labels in image datasets and the noises in industrial data collection are the big problems for deep learning, which would cause overfitting of training model and even performance deterioration. To address this problem, many investigations of learning with noises have been carried out. (1) ***Regularization***: Regularization is a common method to avoid the overfitting, which utilizes an additional penalty term in loss function to control the parameters [36]. The widely used regularization approaches mainly includes: *dropout*, *weight decay*, *adversarial training* and so on [37]. (2) ***Robust loss function***: The concept of robust loss function is similar to regularization, where many different noise-tolerant loss functions were designed to achieve the risk minimization under noisy training data [38, 39, 40]. (3) ***Loss Adjustment***: The last two methods employ fixed loss functions to deal with the noisy labels in dataset, while the method of loss adjustment is able to correct the loss by label transition matrix estimation or reweighted algorithm during the training process [41, 42, 43]. (4) ***Robust model architecture***: This method uses a dedicated layer at the end of neural networks to eliminate the negative effects of noises in dataset. The typical approach in this category is the design of noise adaptation layer, which could estimate the noise probability distribution during the training and further make the neural networks adaptive to noises [44, 45]. (5) ***Meta Learning***: Meta learning is a novel deep learning philosophy, whose core idea is learning to learn. Based on such method, the neural networks could learn to update the loss function under noisy

labeled data [46, 47]. (6) ***Another methods***: There are still many other techniques proposed to deal with the deep learning with noisy training data, such as *sample selection*, *semi-supervised learning* and so on [48, 49]. All of them are developed according to the different tasks and constraints.

Considering the characteristics of thermal modeling in DCs and the requirements of constrained weights in ICNN, the reweighting example algorithm in [27] is adopted to implement robust learning in this work. Specially, the reweighting example algorithm needn't to change the model structure and could be applied for any type of deep neural networks. Such distinguishing features make the algorithm has great advantage in ICNN training.

Now we define $f(x, \theta)$ as the neural network, $L(\tilde{y}, y)$ as the loss function, (x, y) as the inputs-outputs pair, $\tilde{y}$ as the calculation of neural network, $\{(x_i, y_i)|1 \leq i \leq N\}$ as the training dataset, and $\{(x_j^v, y_j^v)|1 \leq i \leq M\}$ as the clean and trusted validation dataset. In the sets and functions, θ denote the parameters of the neural network, N is number of training data, M is number of clean validation data, and the number of clean data is far less than the number of training data ($M \ll N$). The clean validation dataset is contained in the training dataset, and would be used in the training process.

In traditional training of neural networks, the parameters θ could be derived by minimizing the loss function and they are computed equally:

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^{N} L(\tilde{y}_i, y_i) = \arg \min_{\theta} \sum_{i=1}^{N} l_i(\theta), \tag{5.4.5}$$

where θ^* stands for the optimal values of θ, and $l_i(\theta)$ is the transformation of $L(\tilde{y}_i, y_i)$ with respect to θ.

In order to reduce the effects of noisy training data, we need to decrease the proportion of noisy examples and increase the the proportion of clean examples in the loss function. Thus, a weighted loss function is proposed here to reweight all the input examples. Then the equation of loss minimization could be expressed as below:

$$\theta^*(\beta) = \arg\ \min_{\theta} \sum_{i=1}^{N} \beta_i l_i(\theta), \tag{5.4.6}$$

where β is defined as the training weights in the loss function. At beginning, initial values are assigned to β, and then it is optimized based on the following function and clean validation data in each iteration. The optimization of training weight β based on clean validation data is the procedure of figuring out the noisy data in the training dataset and reassigning weights.

$$\beta^* = \arg\ \min_{\beta} \sum_{i=1}^{M} l_i^v(\theta^*(\beta)), \tag{5.4.7}$$

where β^* stands for the optimal value of β, and $l_i^v(\cdot)$ means the loss function is transformed based on the clean validation data. It is noteworthy that the weights β should be keep non-negative, or the minimization with negative training loss cannot derive optimal results.

At last, the sum of all the weights should be equal to one. Here we could utilize the following equation to normalize the weights of all samples.

$$\tilde{\beta}_i^* = \frac{\beta_i^*}{\sum \beta_i^* + \delta(\sum \beta_i^*)}, \tag{5.4.8}$$

where

$$\delta(\alpha) = \begin{cases} 1, & \text{if } \alpha = 0 \\ 0, & \text{if } \alpha \neq 0 \end{cases}. \tag{5.4.9}$$

The use of $\delta(\alpha)$ is to avoid the extreme case that all the weights are equal to zero.

In conclusion, based on the two optimization loops, the optimal weights β^* and layer parameters θ^* could be calculated alternately in each iteration. Step on the layer parameters θ^* calculated by mini-batch training examples, the training examples which are similar to the clean validation set would be up-weighted through the Eq. (5.4.7), and the examples which are noisy and dissimilar to the trusted validation set would be down-weighted.

Remark: The authors in [27] have developed a method of online approximation to simplify the optimization loops, and speed the model training, while we don't adopt this method in this work. First, the amount of industrial data in DCs isn't as huge as the image set, so the time-saving algorithm is not urgent need for thermal modeling. Secondly, the transformation of online approximation might casuse unstable behaviors and increase the difficulty of constraining layer weights during the ICNN training.

5.5 Experimental-data-based Simulation results

All the training and validation data is collected in a experimental single-rack DC with rack-based cooling system. As is shown in Figure 5.7, the servers are placed in an enclosed rack and managed individually, so we just need to investigate the techniques of modeling or control based on a single-rack DC and expand them to the whole DC. Each rack includes nearly 30 servers, one cooling system, several temperature sensors,

humidity sensors, pipes and cables. The micro-controller used in the cooling unit is a specifically designed circuit board mainly comprising Raspberry Pi Zero, Arduino Nano, relays and some cables, where the Arduino Nano is used to collect sensors data and transmit control signal and Raspberry Pi Zero is utilized to implement the control algorithm.

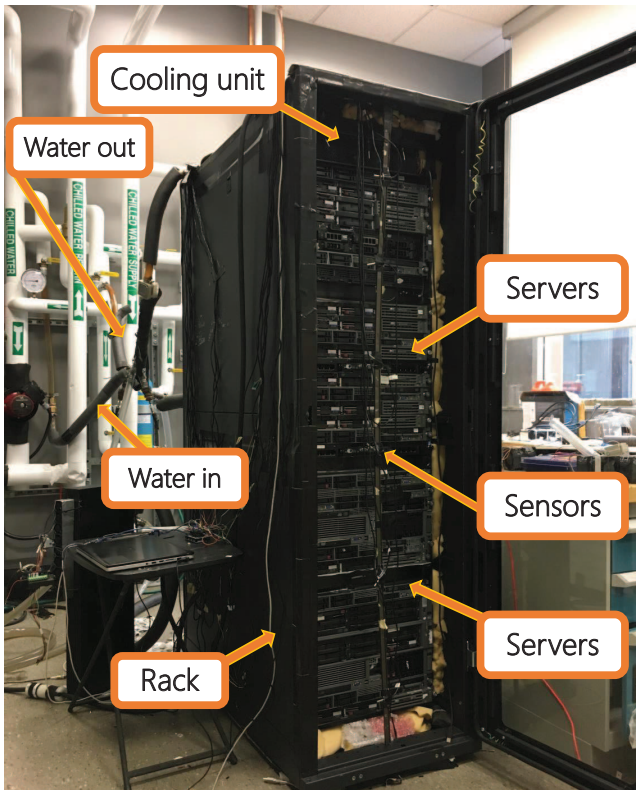

Figure 5.7: Experimental DC with rack-based cooling architecture.

In order to build an accurate enough neural network, the experiments covering almost all working conditions are conducted to collect the training data. The experiments of data collection includes 10 working cycles with respect to different input water flow, air flow, and IT work load. The experiments cost about 80 hours, and the detailed information of experiments covering all working conditions is represented

in Figure 5.8. Additionally, multiple groups of experiments are also carried out to obtain the data for validation.

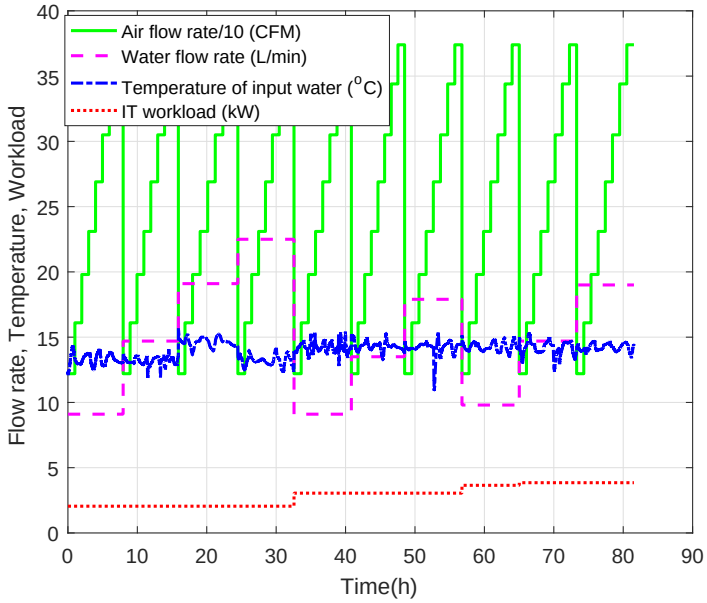

Figure 5.8: Detailed information of experiments covering all working conditions for model training.

We have tried our best to eliminate the environmental disturbances and use the reliable temperature sensors during the experiments, thus, the collected training and validation data is regarded as clean and trusted in this work. As the study in this paper is designed to promote the robustness of ICNN against noisy data, white synthetic noises are added into part of training data randomly. Furthermore, in this work we only investigate the effects of noises in output, and ignore the noises in inputs.

In this section, two simulation cases are carried out to validate our proposed algorithm. In the first case, the IT workload is set as a constant, and the air flow rate and water flow rate are changed every hour. As the IT workload of each regular rack is scheduled as a constant at most of the time, this case simulates the common

working condition in real DCs. The second case is conducted to simulate the specific situation in the operation of DCs, where the IT workload would change due to the heavy network traffic. Therefore, in this case, the air flow rate, water flow rate, and IT workload all change during the experiments. Based on these two simulation cases, the performance of our developed neural network is tested.

5.5.1 Case I: Model validation through the data collected under constant IT workload

In the first case, the experiment of simple working condition is conducted to collect the validation data. During the experiment, the IT workload of whole rack still maintains constant, while air flow and water flow change per hour. The detailed inputs information of the experiment is shown in Figure 5.9. The temperature of input water is considered as the measured disturbance, which cannot be controller through our cooling unit.

The prediction of ICNN trained based on normal method and noisy training data is adopted as the comparison in this work. The detailed comparison of temperature prediction is presented in Figure 5.10, where the red curve denotes the actual temperature data collected from experiment, the black curve stands for the temperature prediction via normal ICNN, and the green curve is the temperature prediction of ICNN trained through example reweighting algorithm. In this paper, the ICNN train through our proposed method is named robust ICNN.

As is seen in Figure 5.10, the overall prediction of robust ICNN is clearly more accurate than the prediction of normal ICNN. Specifically, the performance of robust ICNN at the peak is much better than normal ICNN. Besides, at the last part, the

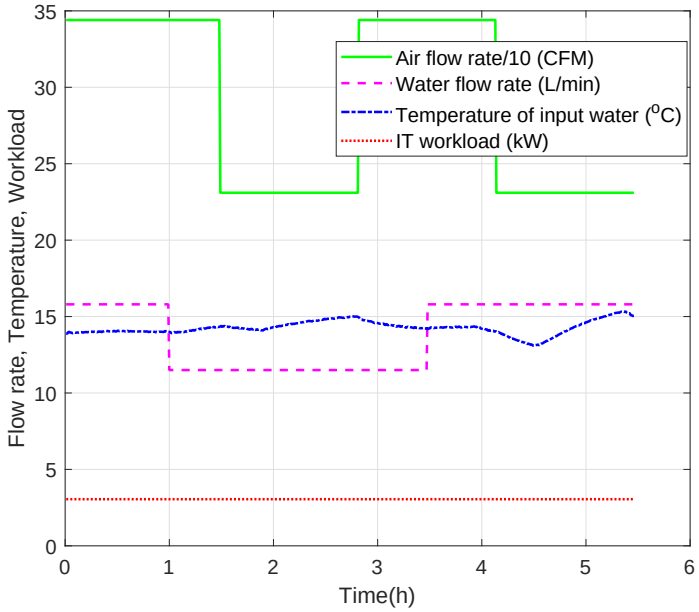

Figure 5.9: Basic inputs information of experiment for case I.

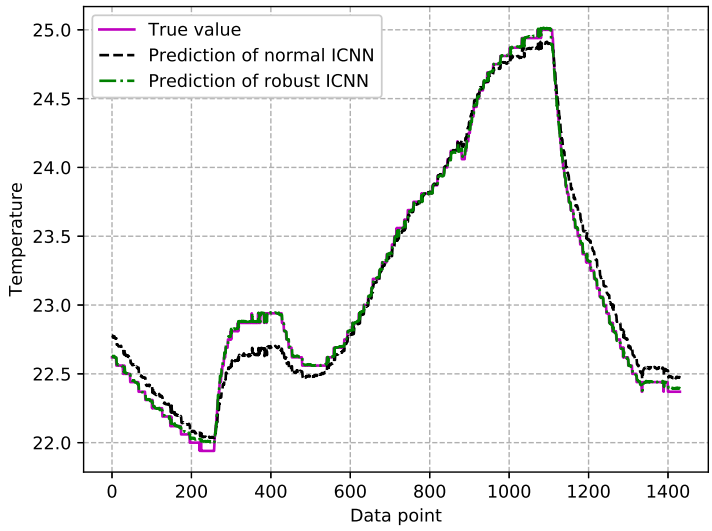

Figure 5.10: The comparison of temperature prediction of normal ICNN and robust ICNN for case I.

prediction of normal ICNN cannot follow the true value very well, while the prediction of robust ICNN could still track the target easily. The results shown in this figure demonstrate that the ICNN just trained through general method is affected by the noises in training data somehow. The ICNN model is over fitted or learns some useless feathers from the noises. On the contrary, the robust ICNN trained by example reweighting algorithm under noisy training data performs very well all the time, which indicates the effectiveness of our proposed technique.

The errors of prediction by two different methods are compared in Figure 5.11. As we can see in the figure, the predicted errors of normal ICNN evidently larger than the errors of robust ICNN, which further prove the superiority of example reweighting algorithm used in this work. Moreover, the predicted errors of robust ICNN still stay within 0.1 degree, which proves the accuracy of our proposed ICNN model is much higher than that of most other data-driven and physics thermal models for DCs.

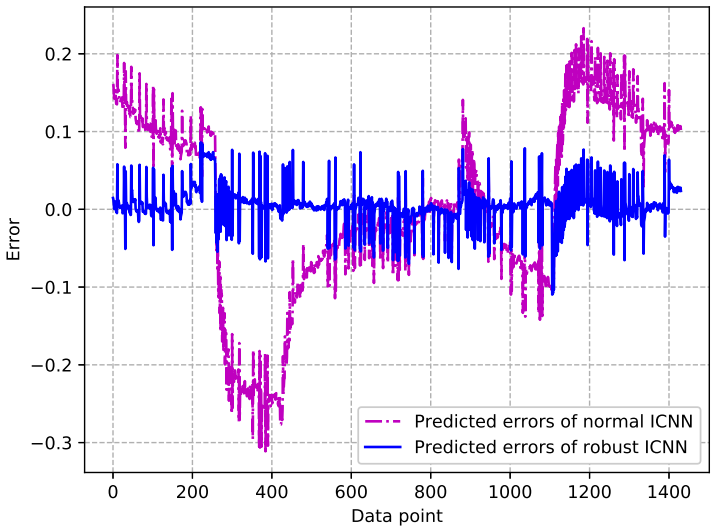

Figure 5.11: The comparison of predicted errors between normal ICNN and robust ICNN in case I.

5.5.2 Case II: Model validation through the data collected under varying IT workload

The experiment in second case is designed to simulate the change of IT workload under exceptional circumstances in the DCs. The varying IT workload would result in more complicated thermal dynamics in the rack and increase the difficult of temperature prediction. The detailed inputs information of the experiment in this case is described in Figure 5.12.

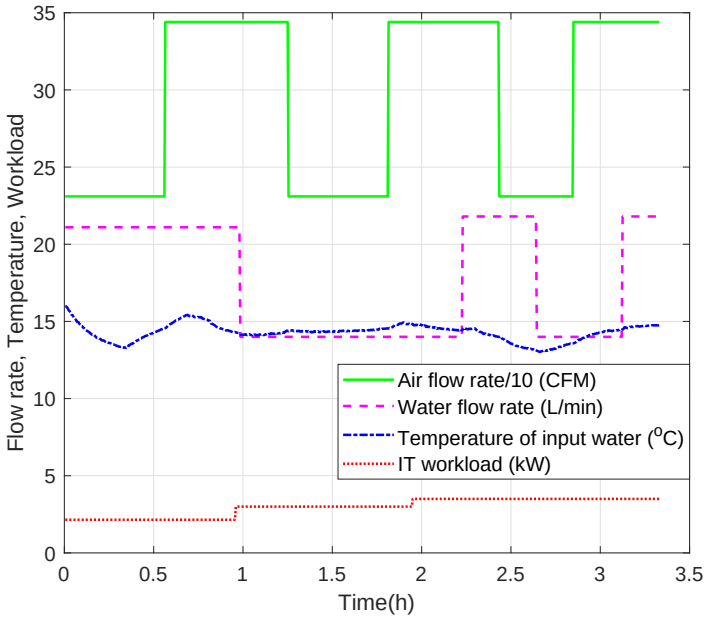

Figure 5.12: Basic inputs information of experiment for case II.

The comparison of temperature prediction between normal ICNN and robust ICNN in this case is depicted in Figure 5.13. The green curve means the prediction of robust ICNN, the black curve represents the the prediction of normal ICNN and the red curve stands for the true value of temperature.

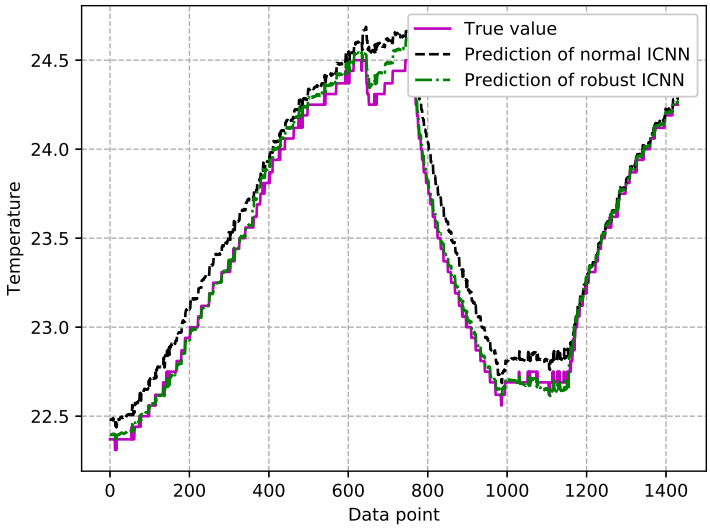

Figure 5.13: The comparison of temperature prediction of normal ICNN and robust ICNN for case II.

In this figure, the prediction of robust ICNN is able to track the true value immediately at the beginning, and the tracking errors are no more than 0.1 degree. However, the prediction of normal ICNN takes a long time to catch the actual value, and the tracking errors are much larger than that of robust ICNN. Nevertheless, the normal ICNN performs really badly at the inflection point, and it always fails to follow the true value at peak as the same as the prediction in the first case. The comparison in this case illustrates that the example reweighting algorithm we adopted in this work could eliminate the effects of noises again, and further avoid overfitting during the training process.

The comparison of predicted errors are expressed in Figure 5.14, where the overall predicted errors of robust ICNN are smaller than that of normal ICNN as well. The performance of robust ICNN in this case is not as good as the performance in case

I, since the varying IT workload in the experiment. However, the predicted errors of robust ICNN don't overstep too many, and the model is still accurate enough.

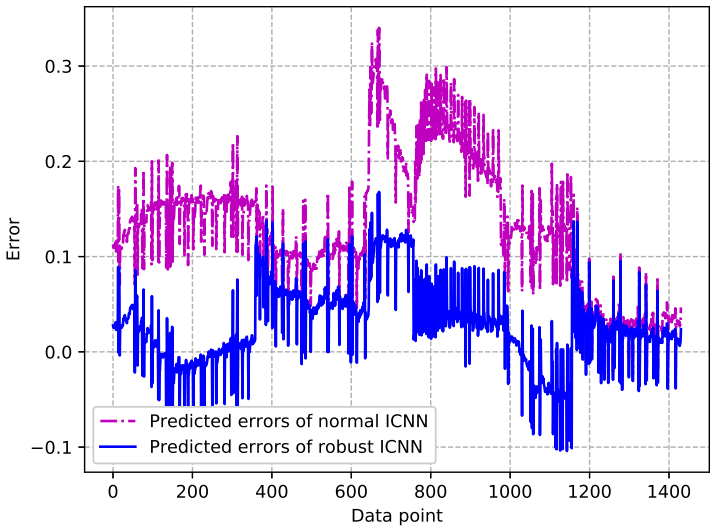

Figure 5.14: The comparison of predicted errors between normal ICNN and robust ICNN in case II.

According to the validations in these two cases, the performance of robust ICNN is tested. Compared with the ICNN model trained through normal method, the performance of ICNN trained via example reweighting algorithm is much better. The validations prove that our developed method could eliminate the negative effects of noisy data during the training process efficiently, and further avoid the overfitting of ICNN. In addition, the small predicted errors also state our designed ICNN is precise enough to describe the thermal dynamics in DCs.

5.6 Conclusion

This work investigates the issue of thermal modeling based on ICNN under noisy training data in DCs. The ICNN is designed for bridging the gap between neural networks and control tractability, and the example reweighting algorithm is employed to avoid the model overfitting under noisy training data. Model validations with different cases of IT workload demonstrate the ICNN trained through example reweighting algorithm could deal with the noises in training data, and thus prevent overfitting. Besides, the simulation results also illustrate the ICNN is much more accurate than most of other data-driven models. In the future, we would study more efficient thermal modeling methods based on deep learning in DCs, and apply the developed ICNN model to DC control soon.

Bibliography

[1] H. Zhang, S. Shao, H. Xu, H. Zou, and C. Tian, "Free cooling of data centers: A review," *Renew. Sustain Energy Rev.*, vol. 35, pp. 171–182, Jul. 2014.

[2] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. Peng, "Green data center with IoT sensing and cloud-assisted smart temperature control system," *Computer Networks*, vol. 101, pp. 104–112, Jun. 2016.

[3] T. Ding, Z. guang He, T. Hao, and Z. Li, "Application of separated heat pipe system in data center cooling," *Appl. Therm. Eng.*, vol. 109, pp. 207–216, Oct. 2016.

[4] T. D. Boucher, D. M. Auslander, C. E. Bash, C. C. Federspiel, and C. D. Patel, "Viability of dynamic cooling control in a data center environment," *J. Electron. Packag.*, vol. 128, no. 2, pp. 137–144, Nov. 2006.

[5] T. Lu, X. Lü, M. Remes, and M. Viljanen, "Investigation of air management and energy performance in a data center in finland: Case study," *Energy and Buildings*, vol. 43, no. 12, pp. 3360–3372, Dec. 2011.

[6] S. MirhoseiniNejad, H. Moazamigoodarzi, G. Badawy, and D. G. Down, "Joint data center cooling and workload management: A thermal-aware approach," *Future Gener. Comput. Syst.*, vol. 104, pp. 174–186, Mar.

[7] H. Cheung, S. Wang, C. Zhuang, and J. Gu, "A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation," *Appl. energy*, vol. 222, pp. 329–342, Jul. 2018.

[8] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renew. Sustain Energy Rev.*, vol. 31, pp. 622–638, Mar. 2014.

[9] H. Moazamigoodarzi, P. J. Tsai, S. Pal, S. Ghosh, and I. K. Puri, "Influence of cooling architecture on data center power consumption," *Energy*, vol. 183, pp. 525–535, Sep. 2019.

[10] L. Parolini, B. Sinopoli, B. H. Krogh, and Z. Wang, "A cyber–physical systems approach to data center modeling and control for energy efficiency," *Proc. IEEE*, vol. 100, no. 1, pp. 254–268, Jan. 2012.

[11] B. Durand-Estebe, C. Le Bot, J. N. Mancos, and E. Arquis, "Data center optimization using PID regulation in CFD simulations," *Energy and Buildings*, vol. 66, pp. 154–164, Nov. 2013.

[12] M. Ogawa, H. Fukuda, H. Kodama, H. Endo, T. Sugimoto, T. Kasajima, and M. Kondo, "Development of a cooling control system for data centers utilizing indirect fresh air based on model predictive control," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 132–137, 2015.

[13] R. Zhou, Z. Wang, C. E. Bash, and A. McReynolds, "Modeling and control for cooling management of data centers with hot aisle containment," in *Proceedings of ASME 2011 International Mechanical Engineering Congress & Exposition*, 2011.

[14] L. Parolini, B. Sinopoli, and B. H. Krogh, "Model predictive control of data centers in the smart grid scenario," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 10 505–10 510, Jan. 2011.

[15] J. Deng, L. Yang, X. Cheng, and W. Liu, "Self-tuning PID-type fuzzy adaptive control for CRAC in datacenters," in *International Conference on Computer and Computing Technologies in Agriculture*, pp. 215–225, 2013.

[16] M. Jonas, R. R. Gilbert, J. Ferguson, G. Varsamopoulos, and S. K. Gupta, "A transient model for data center thermal prediction," in *2012 International Green Computing Conference (IGCC)*, pp. 1–10. IEEE, 2012.

[17] W. A. Abdelmaksoud, H. E. Khalifa, T. Q. Dang, R. R. Schmidt, and M. Iyengar, "Improved CFD modeling of a small data center test cell," in *12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 1–9. IEEE, 2010.

[18] V. A. Tsachouridis and T. Scherer, "Data centre adaptive numerical temperature models," *Trans. Inst. Meas. Control*, vol. 40, no. 6, pp. 1911–1926, Mar. 2018.

[19] H. Moazamigoodarzi, S. Pal, S. Ghosh, and I. K. Puri, "Real-time temperature predictions in IT server enclosures," *Int. J. Heat Mass Transf.*, vol. 127, pp. 890–900, Dec. 2018.

[20] J. Athavale, M. Yoda, and Y. Joshi, "Comparison of data driven modeling approaches for temperature prediction in data centers," *Int. J. Heat Mass Transf.*, vol. 135, pp. 1039–1052, Jun. 2019.

[21] E. Samadiani, Y. Joshi, H. Hamann, M. K. Iyengar, S. Kamalsy, and J. Lacey, "Reduced order thermal modeling of data centers via distributed sensor data," *ASME J. Heat Transfer*, vol. 134, no. 4, Feb. 2012.

[22] R. Ghosh and Y. Joshi, "Error estimation in POD-based dynamic reduced-order thermal modeling of data centers," *Int. J. Heat Mass Transf.*, vol. 57, no. 2, pp. 698–707, Feb. 2013.

[23] K. Jiang, S. Shi, H. Moazanigoodarzi, C. Hu, S. Pal, and F. Yan, "Temperature distribution estimation via data-driven model and adaptive kalman filter in modular data centers," *Proc. IMechE, Part I: J. Systems and Control Engineering*, Apr. 2020.

[24] Z. Song, B. T. Murray, and B. Sammakia, "A dynamic compact thermal model for data center analysis and control using the zonal method and artificial neural networks," *Appl. Therm. Eng.*, vol. 62, no. 1, pp. 48–57, Jan. 2014.

[25] Z. Song, B. T. Murray, and B. Sammakia, "Airflow and temperature distribution optimization in data centers using artificial neural networks," *Int. J. Heat Mass Transf.*, vol. 64, pp. 80–90, Sep. 2013.

[26] S. Tashiro, Y. Nakamura, K. Matsuda, and M. Matsuoka, "Application of convolutional neural network to prediction of temperature distribution in data centers," in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, pp. 656–661. IEEE, 2016.

[27] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *International Conference on Machine Learning*, 2018.

[28] H. Moazamigoodarzi, R. Gupta, S. Pal, P. J. Tsai, S. Ghosh, and I. K. Puri, "Modeling temperature distribution and power consumption in IT server enclosures with row-based cooling architectures," *Appl. Energy*, vol. 261, p. 114355, Mar. 2020.

[29] R. Gupta, S. Asgari, H. Moazamigoodarzi, S. Pal, and I. K. Puri, "Cooling architecture selection for air-cooled data centers by minimizing exergy destruction," *Energy*, p. 117625, Jun. 2020.

[30] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters," *Procedia Comput. Sci.*, vol. 125, pp. 676–682, 2018.

[31] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *International Conference on Machine Learning*, pp. 146–155, 2017.

[32] Y. Chen, Y. Shi, and B. Zhang, "Optimal control via neural networks: A convex approach," in *International Conference on Learning Representations*, 2019.

[33] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[36] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2017.

[37] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in neural information processing systems*, pp. 950–957, 1992.

[38] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in neural information processing systems*, pp. 8778–8788, 2018.

[39] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 322–330, 2019.

[40] Y. Lyu and I. W. Tsang, "Curriculum loss: Robust learning and generalization against label corruption," in *International Conference on Learning Representations*, 2020.

[41] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017.

[42] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in *Advances in neural information processing systems*, pp. 10 456–10 465, 2018.

[43] H.-S. Chang, E. Learned-Miller, and A. McCallum, "Active bias: Training more accurate neural networks by emphasizing high variance samples," in *Advances in Neural Information Processing Systems*, pp. 1002–1012, 2017.

[44] I. Jindal, M. Nokleby, and X. Chen, "Learning deep networks from noisy labels with dropout regularization," in *IEEE 16th International Conference on Data Mining (ICDM)*, pp. 967–972. IEEE, 2016.

[45] B. Han, J. Yao, G. Niu, M. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama, "Masking: A new perspective of noisy supervision," in *Advances in Neural Information Processing Systems*, pp. 5836–5846, 2018.

[46] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, pp. 1842–1850, 2016.

[47] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*, 2017.

[48] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International Conference on Machine Learning*, pp. 2304–2313. PMLR, 2018.

[49] Y. Yan, Z. Xu, I. W. Tsang, G. Long, and Y. Yang, "Robust semi-supervised learning through label aggregation," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Chapter 6

Conclusions and future work

6.1 Conclusions

Thermal management is one of the most important topics in DCs, which is closely related to the safe operation and energy-efficiency of ITEs. This thesis mainly investigate the application of data-driven techniques for thermal management in DCs. Different data-driven models are applied for thermal modeling, temperature distribution estimation and temperature control in DCs with rack-based cooling architecture.

To reduce the cost and fault diagnosis burden of sensor network, a novel observer combined adaptive Kalman filter and data-driven model is proposed to estimate the temperature distribution in DC. The data-driven model is constructed in the form multiple linear ARX models, and identified by the algorithm if PLS. Besides, FCM is utilized to partition the data set for the training of multiple sub-models. The simulation results demonstrate our proposed observer is accurate enough to replace the physical temperature sensors in DC. Secondly, based on the data-driven model in previous section, a fault tolerant predictive controller is developed in DC. The

data-driven fault tolerant predictive controller is designed to deal with three different types of actuator faults including fans aging, fans failure and valve deactivation. The real experiments in DC prove the excellent performance of the proposed controller.

In order to further improve the data-driven modeling accuracy, a type of ICNN is adopted to represent the thermal dynamics in DC. The ICNN is design to be convex from input to output by constraining the layer weights. Thus, the ICNN could be applied for optimal control with powerful representation capability. In addition, an algorithm of EWC is employed during the training process to overcome the catastrophic forgetting in continual learning. The results of two simulation cases illustrate that the proposed ICNN is not only efficient for multiple task learning, but also robust to noisy training data in sequence learning. Following this idea, an example reweighting algorithm is adopted to improve the robustness of ICNN against noises in training data. The adopted algorithm is easily performed without any additional hyperparameter tuning, and could be applied for any type of DNNs. Finally, two simulation cases demonstrate its effectiveness through experimental data with synthetic noisy data.

In summary, multiple data-driven techniques are employed to address the problems of thermal management in DCs with rack-based cooling architecture. The experimental and simulation results the superiority of data-driven methods. The investigations of data-driven techniques in this thesis may not only be used for thermal management in DCs, but also for other nonlinear industrial systems.

6.2 Future work

The researches in this work demonstrate data-driven techniques perform considerable potential for thermal management and energy management in DCs or other industrial systems. Therefore, some future works are recommended with respect to data-driven methods and managements in DCs.

1. Implementing ICNN-based predictive control to regulate the temperature in DC with rack-based or row-based cooling architecture.

2. Employing data-driven methods to combine the thermal and energy models together, and exploring the joint control method in DCs.

3. Developing reinforcement learning for thermal and energy management in DCs, validating the control performance and comparing with ICNN-based control.

4. Investigating the application of data-driven techniques for fault diagnosis in DCs.

5. Developing efficient DNNs with lower computational load to accelerate the application of DNN-based controller in industry.