Intelligent Frost Prediction and Active Protection Cyber-physical Systems

in

the Agricultural Sector

A thesis submitted in fulfillment
of the requirements for the degree of

Doctor of Philosophy

by

Ian Zhou

School of Electrical and Data Engineering

Faculty of Engineering and Information Technology

University of Technology Sydney

NSW 2007, Australia

June 2023

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Ian Zhou declare that this thesis, is submitted in fulfillment of the requirements for the award of the degree of Doctor of Philosophy, in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

This research is also supported by Food Agility CRC Ltd, funded under the Commonwealth Government CRC Program. The CRC Program supports industry-led collaboration between industry, researchers, and the community.

Signed:

Production Note:
Signature removed prior to publication.

Ian Zhou
June 02, 2023

ABSTRACT

Frost damage in broadacre cropping and horticulture (including viticulture) results in substantial economic losses to producers and may also disrupt associated product value chains. Frost risk windows are changing in timing, frequency, and duration. Faced with the increasing cost of mitigation infrastructure and competition for resources (e.g., water and energy), multi-peril insurance, and the need for supply chain certainty, producers are under pressure to innovate in order to manage and mitigate risk. Frost protection systems are cyber-physical systems comprising sensors (event detection), intelligence (prediction), and actuators (active protection methods). These systems are an important tool and cost factor in fighting against frost. Therefore, there is a need to improve the efficiency and effectiveness of frost protection cyber-physical systems.

This study adopts the Internet of Things 2.0 architectures and emphasizes the dimensions of machine learning intelligence, scalability, interoperability, and user-friendliness in frost protection systems. This research also improves on the limitations of existing frost protection systems and the prediction methods that control the systems. The limitations are historical data dependence, low prediction temporal resolution, non-real-time response, and low fault tolerance. In response to the limitations of low prediction temporal resolution and non-real-time response, the existing 12-24 hours prediction methods are extended by artificial neural networks and recurrent neural networks for near real-time frost prediction. A minute-wise regression model to predict the next hour minimum temperature is proposed. The minute-wise model further highlights the system dependence on local historical data and sensors. To decouple this dependence, a spatial interpolation-based frost prediction system is implemented. This model-based system only requires data from existing weather stations to predict frost at any new sites. Combining the results of the previous two models, a cyber-physical system framework is proposed to improve the system fault tolerance. This framework is a modular design with the local data-based system as the primary predictor and the model-based system as both a secondary predictor and system stopping mechanism. The result shows improvements in operational cost compared to traditional methods. The final contribution is the improvement of energy efficiency on edge-based prediction. A spatially generalized model with a smaller prediction window is constructed and deployed on a LoRa transmission node. The proposed system not only improves energy efficiency, it also reduces the false positive rate.

DEDICATION

In memory of my grandmother Cui Qin Weng.

ACKNOWLEDGMENTS

I would like to express my gratitude to my principal supervisor, A/Prof. Justin Lipman, who guided me throughout this journey of research and supported me with great patience. Without his continuous inspiration to me, this thesis would not have been fulfilled.
I am truly thankful to my co-supervisor, Prof. Mehran Abolhasan, for leading me into the gateway of research and his ongoing support with his knowledge in the academic world.
I am deeply grateful to my co-supervisor, Dr. Negin Shariati, who gave me the opportunity to be on this path of research and supported me with her rich knowledge.
I would like to extend my sincere thanks to my colleague, Dr. Imran Makhdoom, for his positive influence on my style of research.
I would also acknowledge the University of Technology Sydney for providing me the opportunity to pursue my Ph.D. degree and the administration support from the university staff.
Finally, I would like to thank my family and friends for the invaluable encouragement through my studies.

List of Publications

Journal Papers

J1. I. Zhou, I. Makhdoom, N. Shariati, M.A. Raza, R. Keshavarz, J. Lipman, M. Abolhasan and A. Jamalipour, "Internet of Things 2.0: Concepts, Applications, and Future Directions," in *IEEE Access*, vol. 9, pp. 70961-71012, 2021, doi: 10.1109/ACCESS.2021.3078549. **Thesis Chapter 2**

J2. I. Zhou, J. Lipman, M. Abolhasan, N. Shariati and D. W. Lamb, "Frost Monitoring Cyber–Physical System: A Survey on Prediction and Active Protection Methods," in *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6514-6527, July 2020, doi: 10.1109/JIOT.2020.2972936. **Thesis Chapter 2**

J3. I. Zhou, J. Lipman, M. Abolhasan and N. Shariati, "Minute-wise Frost Prediction: An Approach of Recurrent Neural Networks," in *Array*, vol. 14, pp. 100158, April 2022, doi: 10.1016/j.array.2022.100158. **Thesis Chapter 3**

J4. I. Zhou, J. Lipman, M. Abolhasan and N. Shariati, "Intelligent Spatial Interpolation-based Frost Prediction Methodology using Artificial Neural Networks with Limited Local Data," **In press** in *Environmental Modelling & Software*, doi: 10.1016/j.envsoft.2023.105724. **Thesis Chapter 4**

J5. I. Zhou, J. Lipman, M. Abolhasan and N. Shariati, "A Frost Forecasting Framework Featuring Spatial Prediction-based Stopping Mechanism," Submitted to *IEEE Internet of Things Journal*. **Thesis Chapter 5**

J6. I. Zhou, J. Lipman, M. Abolhasan and N. Shariati, "Edge-based Spatially Generalized Frost Prediction with Neural Networks," Submitted to *IEEE Internet of Things Journal*. **Thesis Chapter 6**

Statement of Thesis by Compilation

Title of Published/Publishable Content	Author list	Student's Contribution	Status
Internet of Things 2.0: Concepts, Applications, and Future Directions (**Chapter 2**)	Ian Zhou Imran Makhdoom Negin Shariati Muhammad Ahmad Raza Rasool Keshavarz Justin Lipman Mehran Abolhasan Abbas Jamalipour	The student contribution in the published version of this paper are contents of Sections 1-4, 6, 10-12, 14, Subsections 13.A, B, D, F, G, 13.C.2. Only the above sections are included in the thesis. The student also took a major part in conceptualization, final editing and formatting of the published paper. The estimated student's contribution is **70%** of the published paper. The co-authors drafted other sections of the published paper and also contributed to the validation of the content.	Published in IEEE Access
Frost monitoring cyber-physical system: a survey on prediction and active protection methods (**Chapter 2**)	Ian Zhou Justin Lipman Mehran Abolhasan Negin Shariati David William Lamb	The student's contribution is about **85%**. The co-authors of this paper provided correction and feedbacks to the drafted content. They are also involved in minor editing of the content.	Published in IEEE Internet of Things Journal
Minute-wise Frost Prediction: An Approach of Recurrent Neural Networks (**Chapter 3**)	Ian Zhou Justin Lipman Mehran Abolhasan Negin Shariati	The student's contribution is about **90%**. This includes paper drafting, model training and testing. The co-authors of this paper provided correction and feedbacks to the drafted content.	Published in Elsevier Array
Intelligent Spatial Interpolation-based Frost Prediction Methodology using Artificial Neural Networks with Limited Local Data (**Chapter 4**)	Ian Zhou Justin Lipman Mehran Abolhasan Negin Shariati	The student's contribution is about **90%**. This includes paper drafting, model training and testing. The co-authors of this paper provided correction and feedbacks to the drafted content.	In Press in Environmental Modelling and Software
A Frost Forecasting Framework Featuring Spatial Prediction-based Stopping Mechanism (**Chapter 5**)	Ian Zhou Justin Lipman Mehran Abolhasan Negin Shariati	The student's contribution is about **90%**. This includes paper drafting, model training and testing. The co-authors of this paper provided correction and feedbacks to the drafted content.	Submitted to IEEE Internet of Things Journal
Edge-based Spatially Generalized Frost Forecasting with Neural Networks (**Chapter 6**)	Ian Zhou Justin Lipman Mehran Abolhasan Negin Shariati	The student's contribution is about **90%**. This includes paper drafting, hardware setup, model training and testing. The co-authors of this paper provided correction and feedbacks to the drafted content.	Submitted to IEEE Internet of Things Journal

	Signature	Date
Co-Author: Justin Lipman	Production Note: Signature removed prior to publication.	06/06/2023
Co-Author: Mehran Abolhasan	Production Note: Signature removed prior to publication.	06/06/2023
Co-Author: Negin Shariati	Production Note: Signature removed prior to publication.	06/06/2023
Co-Author: Abbas Jamalipour	Production Note: Signature removed prior to publication.	6/06/2023
Co-Author: Rasool Keshavarz	Production Note: Signature removed prior to publication.	07/06/2023
Co-Author: David William Lamb	David W. Lamb (dig.sig)	7 June 2023
Co-Author: Imran Makhdoom	Production Note: Signature removed prior to publication.	7/6/2023
Co-Author: Muhammad Ahmad Raza	Production Note: Signature removed prior to publication.	07-Jun-2023
Student: Ian Zhou	Production Note: Signature removed prior to publication.	07/06/2023

CONTENTS

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Challenges . . . 2

1.3 Research Questions and Objectives . . . 2

1.4 Deliverables . . . 3

1.5 Stakeholders . . . 3

1.6 Research Methodology . . . 4

1.7 Thesis Organization . . . 6

2 Literature Review **7**

2.1 Technologies and Concepts underlying IoT 2.0 . . . 8

2.1.1 5G . . . 10

2.1.2 Tactile Internet . . . 11

2.1.3 Edge Computing . . . 11

2.1.4 Industry 4.0 . . . 12

2.1.5 Machine Learning . . . 12

2.2 IoT Architectures . . . 15

2.2.1 Cyber-Physical Systems . . . 17

2.3 Machine Learning Intelligence . . . 19

2.3.1 Machine Learning Algorithms . . . 19

2.3.2 Physical Layer Applications . . . 34

2.3.3 Network Layer Applications . . . 35

2.3.4 Edge Computing Applications . . . 38

2.3.5 Edge-Cloud Collaboration . . . 42

2.4 IoT Scalability . . . 44
2.4.1 SDN Induced Scalability . . . 46
2.5 IoT Interoperability: Interoperability Between Standards . . . 49
2.6 User friendly IoT . . . 50
2.7 Concepts of Frost . . . 52
2.8 Frost Prediction Methods . . . 53
2.8.1 Classification Methods . . . 56
2.8.2 Regression Methods . . . 57
2.9 Frost Protection Methods . . . 61
2.9.1 Passive Frost Protection Methods . . . 61
2.9.2 Active Frost Protection Methods . . . 61
2.10 Integrated Frost Prediction and Active Protection Systems . . . 64
2.11 IoT Communication Protocols for Frost Protection Applications . . . 65
2.12 Current Limitations and Future Directions of Frost Systems . . . 68
2.13 Summary . . . 70
2.14 Review on Post-thesis Frost Prediction . . . 71

3 Minute-wise Frost Prediction: An Approach of Recurrent Neural Networks 75
3.1 Introduction . . . 75
3.1.1 Related work . . . 76
3.2 Methodology . . . 77
3.2.1 Study Area . . . 77
3.2.2 Data Preprocessing . . . 79
3.2.3 Model Construction and Testing . . . 81
3.2.4 Experiments . . . 82
3.3 Results and Discussions . . . 83
3.3.1 Model Error . . . 83
3.3.2 Effect of Sequence Length on Model Error . . . 85
3.3.3 Effect of Sequence Length on Processing Time . . . 86
3.3.4 Limitations and Open Challenges . . . 89
3.4 Summary . . . 91

4 Intelligent Spatial Interpolation-based Frost Prediction Methodology using Artificial Neural Networks with Limited Local Data 93

4.1 Introduction . . . 93

4.1.1 Related Work . . . 94

4.2 Methodology . . . 95

4.2.1 Data Sources . . . 95

4.2.2 Data Preprocessing . . . 96

4.2.3 Model Types . . . 99

4.2.4 Experiments . . . 100

4.3 Results and Discussions . . . 101

4.3.1 Effect of Different Fold Training Datasets . . . 101

4.3.2 Model Accuracy . . . 104

4.3.3 Effect of the Number of Available Weather Stations . . . 104

4.3.4 Comparing Proposed Data Aggregation Methods with Traditional Methods 107

4.3.5 Limitations and Open Challenges . . . 109

4.4 Summary . . . 111

5 A Frost Forecasting Framework Featuring Spatial Prediction-based Stopping Mechanism 113

5.1 Introduction . . . 113

5.1.1 Related Work . . . 114

5.2 Methodology . . . 116

5.2.1 Proposed Framework Structure . . . 116

5.2.2 Data Sources . . . 118

5.2.3 Experiments . . . 119

5.3 Results and Discussions . . . 120

5.3.1 Accuracy of Different Merging Methods . . . 120

5.3.2 Frost Detection with Different Transmission Periods . . . 121

5.3.3 Operation Time of Protection Equipment . . . 123

5.3.4 Limitations and Open Challenges . . . 125

5.4 Summary . . . 127

6 Edge-based Spatially Generalized Frost Prediction with Neural Networks 129

6.1 Introduction . . . 129

6.1.1 Related Work . . . 130

6.2 Methodology . . . 131

6.2.1 Model Construction . . . 131

6.2.2 Testing Node . . . 132

6.2.3 Experiments . . . 132

6.3 Results and Discussions . . . 135

6.3.1 Daily Energy Consumption of the Testing Node . . . 135

6.3.2 Frost Detection with Different Model Types . . . 137

6.3.3 Composition of Equipment Operation Time . . . 138

6.3.4 Frost Detection with Reduced Prediction Windows . . . 139

6.3.5 Limitations and Open Challenges . . . 141

6.4 Summary . . . 143

7 Conclusions and Future Work 145

7.1 Summary of Thesis Chapters . . . 146

7.2 Future Work . . . 147

7.2.1 Increase the Accuracy of Weak Predictors . . . 148

7.2.2 Define and Propose Important Requirements of Standard Datasets . . . 148

7.2.3 Study the Frost Tolerance of Important and Vulnerable Plants . . . 148

Bibliography 149

A Weather Station Location and Fold Distribution 177

B Raster Maps Generated from Models Created by Different Weather Stations 181

List of Figures

1.1 Objectives, Research Questions, and Deliverables. 2
1.2 Research Methodology. 5

2.1 IoT 2.0 Concepts. 9
2.2 Major tasks of edge computing. [1] 13
2.3 Layered Conventional IoT Architecture. [2] 14
2.4 Recent IoT Layered Architectures. 16
2.5 Basic CPS Architecture. [3] 18
2.6 Support vector machine. 20
2.7 One-dimensional input linear regression. 22
2.8 KNN inference with three neighbors. 23
2.9 Sample Feedforward Neural Network Architecture. 27
2.10 General Convolutional Neural Network Architecture. 27
2.11 2D Convolution Filtering. 28
2.12 RNN Layer at time t. 29
2.13 Agent-Environment Relationship. [4] 33
2.14 SDN Architecture. [5] 47
2.15 NFV Architecture. [6] 48
2.16 Software defined NFV Architecture. [7] 48

3.1 Weather Stations with ID 78
3.2 Data Preprocessing Steps. 80
3.3 Average MSE Tested with Current and Next Year Datasets 84
3.4 Average MSE Tested with Current Year Datasets for Different Sequence Lengths 85

3.5 Average MSE Tested with Next Year Datasets for Different Sequence Lengths . . 86
3.6 Average Training Time per Epoch for Different Sequence Lengths 87
3.7 Average Inference Time per Input for Different Sequence Lengths 88

4.1 Training and Testing with Five-fold Validation. 97
4.2 Distribution of Station Folds on NSW and ACT DEM Map. 98
4.3 Raster Maps from Models Using Weather Station 63291 as Climate Data Source Trained with the Datasets from Folds 1–4. 101
4.4 Raster Maps from Averaged Results per Fold. 102
4.5 Absolute Error Map from Averaged Results Between Folds. 103
4.6 RMSE of Individual, Averaged, Weighted Averaged Models Obtained from Year 2017 Testing Datasets. 104
4.7 RMSE of Averaged and Weighted Averaged Models with 10–60 Available Stations Obtained from Year 2018 Testing Datasets. 105
4.8 True Positive Rate and False Discovery Rate of Averaged, Weighted Averaged, and Weighted Voting Models with 10–60 Available Stations Obtained from Year 2018 Testing Datasets. 106
4.9 RMSE of Averaged and Weighted Averaged Models with 1–10 Available Stations Obtained from Year 2018 Testing Datasets. 106
4.10 True Positive Rate and False Discovery Rate of Averaged, Weighted Averaged, and Weighted Voting Models with 1–10 Available Stations Obtained from Year 2018 Testing Datasets. 107
4.11 RMSE of Averaged, Weighted Averaged, IDW, and OK Models on 2018 Fold 0 Data. 108
4.12 True Positive Rate and False Discovery Rate of Averaged, Weighted Averaged, Weighted Voting, IDW, and OK Models on 2018 Fold 0 Data. 109

5.1 Experimental Implementation of the Proposed Prediction Model. 117
5.2 Mean RMSE of the Particle Filter Merging Method with One-minute Transmission Period and Variable Number of Particles. 121
5.3 Per Site Comparison between the True Positive Rates of the Kalman Filter results and the Baseline with Transmission Periods of 1, 5, 10, 15 minutes. 124
5.4 Mean Operation Hours and True Positive Rate of the Kalman Filter Merging Method with Variable Transmission Periods. 125

6.1 Concept Diagram of the Experiments. 134

6.2 Simulated and Measured Daily Energy Consumption using local or generalized models with variable transmission/inference periods. 136

6.3 Protection Equipment Mean Operation Time Compositions of the Baseline and Generalized Models with Variable Periods. 140

6.4 Protection Equipment Mean Operation Time Compositions of the Baseline and Generalized Models with Variable Periods and Prediction Windows. 142

B.1 Raster Maps from Models Using Weather Station 58212 as Climate Data Source Trained with the Datasets from Folds 0, 1, 3, and 4. 182

B.2 Raster Maps from Models Using Weather Station 72160 as Climate Data Source Trained with the Datasets from Folds 0, 1, 2, and 4. 182

B.3 Raster Maps from Models Using Weather Station 67119 as Climate Data Source Trained with the Datasets from Folds 0–3. 183

B.4 Raster Maps from Weighted Averaged Results per Fold. 183

B.5 Absolute Error Map from Weighted Averaged Results Between Folds. 184

List of Tables

2.1 Advantages of WSN in Agriculture. [8] . . . 17
2.2 Network Applications and Related Machine Learning Algorithms. [9, 10] . . . 36
2.3 Machine Learning Applications in Self-configuration. [11] . . . 37
2.4 Machine Learning Applications in Self-optimization. [11] . . . 38
2.5 Machine Learning Applications in Self-healing. [11] . . . 38
2.6 Machine Learning Edge Computing Hardware. . . . 40
2.7 Motivation of Edge Computing. . . . 42
2.8 Applications Involving Edge-Cloud Collaboration. . . . 44
2.9 Types of Scalabilty. . . . 46
2.10 Types of IoT Interoperability. [12] . . . 51
2.11 Environmental Characteristics of Radiation and Advection Frosts. [13] . . . 53
2.12 Frost Classification Prediction Methods. . . . 54
2.13 Frost Classification Prediction Methods (Cont.). . . . 55
2.14 Frost Regression Prediction Methods. . . . 58
2.15 Frost Regression Prediction Methods (Cont.). . . . 59
2.16 Common Passive Frost Protection Methods. [13–16] . . . 62
2.17 Common Active Frost Protection Methods. [13–17] . . . 63
2.18 Works on Active Frost Protection Methods. . . . 63
2.19 Works on Automated Frost Protection Systems. . . . 65
2.20 IIoT Requirements. [18] . . . 66
2.21 Comparison of IoT Communication Protocol Energy Consumption and Communication Range. [19–22] . . . 67
2.22 Costs of LPWAN Implementation. [22] . . . 68

2.23 Post-thesis Frost Classification Prediction Methods. 73
2.24 Post-thesis Frost Regression Prediction Methods. 74

3.1 P-values of Comparing Average MSEs for Current Year Datasets 84
3.2 P-values of Comparing Average MSEs for Next Year Datasets 84
3.3 P-values of Comparing Average MSEs between ANN and RNN-based Models with Different Model Sequence Lengths (Current Year) 86
3.4 P-values of Comparing Average MSEs between ANN and RNN-based Models with Different Model Sequence Lengths (Next Year) 87
3.5 P-values of Comparing Average Training Time per Epoch between ANN and RNN-based Models with Different Model Sequence Lengths 88
3.6 P-values of Comparing Average Inference Time per Input between ANN and RNN-based Models with Different Model Sequence Lengths 89

4.1 Features and Predictands of Preprocessed Datasets 98
4.2 Adjustable Weights for Different Folds 100

5.1 Recent Works on Frost Protection Systems. 115
5.2 Climate and Location Data Sources from Different Organizations 119
5.3 Experiment Conditions 120
5.4 Mean RMSE of the Sole LPM and Different Result Merging Methods with One-minute Transmission Period and without Assumed Influence from Frost Protection Equipment. 120
5.5 Baseline Results. 121
5.6 Mean RMSE ($^{\circ}C$) of Different Result Merging Methods with Variable Transmission Periods and Assumed Influence from Frost Protection Equipment. 122
5.7 Mean True Positive Rate of Different Result Merging Methods with Variable Transmission Periods and Assumed Influence from Frost Protection Equipment (Bold values are better than the baseline). 122
5.8 Mean False Discovery Rate of Different Result Merging Methods with Variable Transmission Periods and Assumed Influence from Frost Protection Equipment (All values are better than the baseline). 123
5.9 Mean Operation Hours of Different Result Merging Methods with Variable Transmission Periods and Assumed Influence from Frost Protection Equipment (Bold values are better than the baseline). 123

6.1 Sensor Communication and Decision-making Methods Applied by Recent Frost Protection Systems. . . . 131
6.2 DFRobot SEN0137 Sensor Specifications [23]. . . . 133
6.3 Fixed LoRa Transmission Parameters . . . 133
6.4 NS3 LoRa Daily Energy Consumption Simulation Parameters . . . 133
6.5 Daily Energy Consumption Experiments Settings . . . 135
6.6 Measured Daily Consumption (mWh) with Variable Transmission/Inference Periods (Bold values are the lowest in each column). . . . 137
6.7 Mean RMSE of the Baseline and Generalized Models. . . . 137
6.8 Mean True Positive Rate of the Baseline and Generalized Models with Variable Periods (Bold values are the highest in each column). . . . 138
6.9 Mean False Discovery Rate of the Baseline and Generalized Models with Variable Periods (Bold values are the lowest in each column). . . . 138
6.10 Mean Operation Hours of the Protection Equipment Controlled with the Baseline and Generalized Models with Variable Periods (Bold values are the lowest in each column). . . . 138
6.11 Mean RMSE ($^{\circ}C$) of the Baseline and Generalized Models With Variable Prediction Windows. . . . 139
6.12 Mean True Positive Rate of the Baseline and Generalized Models with Variable Periods and Prediction Windows. . . . 141
6.13 Mean False Discovery Rate of the Baseline and Generalized Models with Variable Periods and Prediction Windows. . . . 141

A.1 Weather Station Location and Fold Distribution (Bold Stations Appeared in Chapter 3) . . . 178
A.2 Weather Station Location and Fold Distribution (Bold Stations Appeared in Chapter 3)(Cont.) . . . 179
A.3 Weather Station Location and Fold Distribution (Bold Stations Appeared in Chapter 3)(Cont.) . . . 180

List of Acronyms

5G	Fifth-generation of Mobile Telecommunications Technology
ACT	Australian Capital Territory
AE	Autoencoders
AI	Artificial Intelligence
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
BN	Bayesian Network
BOM	Bureau of Meteorology
CF	Collaborative Filtering
CNN	Convolutional Neural Network
CPS	Cyber-Physical System
CPT	Conditional Probability Table
D2D	Device to Device
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DEM	Digital Elevation Model
DM	Diffusion Map
DT	Decision Tree
EM	Expectation Maximization
ETSI	European Telecommunications Standards Institute
FFNN	Feedforward Neural Network
GMM	Gaussian Mixture Modeling
GPR	Gaussian Process Regression
GRU	Gated Recurrent Unit
H2H	Human to Human
HCA	Hierarchical Clustering Analysis
HNN	Hopfield Neural Network
HVAC	Heating, Ventilation, and Air Conditioning
IaaS	Infrastructure as a Service
IDW	Inverse Distance Weighting
IIoT	Industrial Internet of Things

iTaaS	Internet of Things as a Service
ITU	International Telecommunication Union
IoT	Internet of Things
JRC	Joint Research Centre
KBR	Kernel Bayes Rule
KNN	K-Nearest Neighbor
LPM	Local Prediction Module
LPWAN	Low Power Wide Area Network
LSPI	Least-Squares Policy Iteration
LSTM	Long Short-term Memory
M2M	Machine to Machine
MDP	Markov Decision Processes
MDS	MultiDimensional Scaling
MEC	Mobile Edge Computing
MM	Merging Module
MSE	Mean Squared Error
NB	Naïve Bayes Classifier
NDVI	Normalized Difference Vegetation Index
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NSW	New South Wales
OK	Ordinary Kriging
PaaS	Platform as a Service
PCA	Principal Component Analysis
QoE	Quality of Experience
QoS	Quality of Service
RandNN	Random Neural Network
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RPM	Remote Prediction Module
RSSI	Received Signal Strength Indicator
SaaS	Software as a Service
SDN	Software Defined Network
SIS	Selective Inverted Sink
SMOTE	Synthetic Minority Oversampling Technique
SOI	Southern Oscillation Index
SOM	Self-Organizing Map
SON	Self-Organizing Network
SVM	Support Vector Machine
SVR	Support Vector Regression

TD	Temporal-Difference
UAV	Unmanned Aerial Vehicle
VNF	Virtualized Network Function
WNFB	Wireless Network Function Virtualization
WSDD	Window Sliding with De-Duplication
WSN	Wireless Sensor Networks

1 Introduction

1.1 Motivation

As the growth of the global population continues, the yield of food production should follow this growth to avoid future food shortages [24]. Although many environmental factors are affecting the yield of food, this research focuses on frost, which can be a cause of food shortages in countries relying on frost intolerant crops [25]. The direct economic loss of lost production through frost damage can be substantial. For example, in 1998, the Wimmera region of Australia lost an estimated 60% of its wheat crop due to frost, costing 200 million AUD [26]. More recent research from 2016 shows that frost events incur a loss of 120 million to 700 million AUD annually from the Australian broadacre agriculture sector [27]. The frost event also adversely impacted on jobs along the supply chain [28].

The global trend of climate change has not only increased the average temperature of the Earth's climate system, but also induced greater instability in weather patterns, which lead towards a higher risk of frost damage [29, 30]. Recent changes in the spatial and temporal characteristics of frost events have been attributed to climate change [27]. Frost prediction models that depend on historical weather data could be challenged. Therefore, real-time active responses provided by Cyber-Physical Systems (CPSs) are required to predict and mitigate the risk of frost damage accurately in the face of these challenges.

1.2 Challenges

Many early designs of frost protection systems referred to IoT as a potential solution. However, the adoption of IoT technology in frost prediction is still primitive. Consequently, the results on the ability of protection are limited. These systems could benefit from modern architectures of IoT. Another challenge comes from the prediction models. Predictions are not real-time. Most prediction models based on machine learning can only perform next day or night predictions. This could be a potential waste of operation costs of protection equipment. The environmental data source is another issue. Existing systems often rely on one local data source. As a result, these systems operate with low fault tolerance. Also, when the protection equipment is operating, the corruption of environmental data source by the protection equipment poses as a challenge.

1.3 Research Questions and Objectives

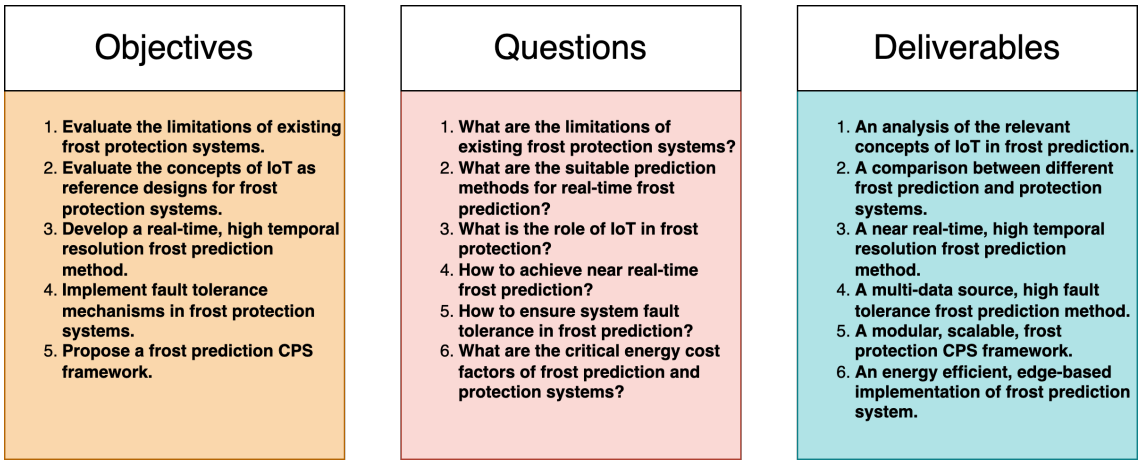


Figure 1.1: Objectives, Research Questions, and Deliverables.

From Figure 1.1, this research aims to increase the temporal resolution and fault tolerance of frost prediction systems by adopting modern IoT concepts. The objectives of this research project are listed as follows:

1. Evaluate the limitations of existing frost protection systems.
2. Evaluate the concepts of IoT as reference designs for frost protection systems.
3. Develop a real-time, high temporal resolution frost prediction method.
4. Implement fault tolerance mechanisms in frost protection systems.
5. Propose a frost protection CPS framework with the following properties:
    - Near real-time prediction and protection.
    - High fault tolerance achieved with multiple data sources.

- Modular framework design.
- Ease of deployment.
- Energy efficient.

The objectives mentioned in the above lead to the following research questions:

1. What are the limitations of existing frost protection systems?
2. What are the suitable prediction methods for real-time frost prediction?
3. What is the role of IoT in frost protection?
4. How to achieve near real-time frost prediction?
5. How to ensure system fault tolerance in frost prediction?
6. What are the critical energy cost factors of frost prediction and protection systems?

1.4 Deliverables

The contributions to achieve the research objectives and answer the research questions are listed as follows:

1. An analysis of the relevant concepts of IoT in frost prediction.
2. A comparison between different frost prediction and protection systems.
3. A near real-time, high temporal resolution frost prediction method.
4. A multi-data source, high fault tolerance frost prediction method.
5. A modular, scalable, frost protection CPS framework.
6. An energy efficient, edge-based implementation of frost prediction system.

1.5 Stakeholders

The deliverables of this research project should influence the academic and industrial sectors of agriculture. The final deliverable includes frost prediction algorithms and a fault-tolerant frost protection CPS. The primary stakeholders affected by this research project are farmers, frost protection equipment manufacturers, and researchers. Farmers are the direct victims of the frost event as a substantial number of crops have been destroyed by frost annually [27]. Hence, farmers can provide accurate primary research results and real-life experiment fields for constructing and validating the proposed system. Since current automated protection systems are incomplete and still in the experimental phase, the farmers rely on frost prediction methods and empirical experiences

to apply protection methods [31]. This would incur excess manual labor costs and operational costs. However, if the proposed system and algorithms can be widely deployed, the operational costs could be reduced. Also, as the system operates autonomously with high fault tolerance, the requirement for manual labor is minimized. Moreover, the damage to crops should be reduced. Overall, with the deployment of the proposed system, the loss incurred by crop damage would be reduced for the farmers.

The frost equipment manufacturers could provide information on existing frost protection equipment and provide recommendations on efficient, effective, and economically feasible equipment. The proposed system intends to automate the usage of these frost protection equipment. Consequently, the manual labor and operational cost of using the equipment decreases. Also, as the automated protection system facilitates the frost protection process, this could drive the development of protection equipment that complies with existing protection systems. A novel industrial communication standard between different components of the protection system could be established as a branch of IoT.

Some deliverables and results of this research project could stimulate other researchers in their future research work. The frost prediction method could provide a baseline for future development to increase model accuracy and resolution. Furthermore, the proposed system could be a reference platform to test novel active frost protection methods. Finally, as mentioned above, this could induce further research on an industry-based communication protocol between frost protection system components.

Frost insurance providers are minor stakeholders. In Australia, there is a limited number of insurances that cover frost damage. A reason for this limitation is the high variability of climate patterns [32]. The proposed frost prediction model should contribute to the accuracy of frost prediction and eventually foster the business opportunity of frost-related insurances. Finally, as current frost events impact all parties along the agricultural supply chain [28], this research project could reduce the economic loss for all these parties.

1.6 Research Methodology

The initial phase of this research started from the challenges defined in Section 1.2. Then, the next phase and the phases after all build upon the limitations and challenges identified in the previous phase. Therefore, later phases of this research are connected and derived by the limitations of earlier phases. From Figure 1.2, the initial phase 0 outlines the relevant concepts and architectures of IoT. Also, the existing frost prediction systems in the literature are evaluated. The analysis of existing frost prediction systems provided the initial limitations and research challenges to start the project. The limitation of low prediction frequencies of the existing frost prediction models is addressed in phase 1. Minute-wise predictions with neural networks became the solution. Also, RNN-based models are assessed as frost prediction models in phase 1. Then, in phase 2, the neural networks are further improved with spatial interpolation techniques to eliminate the requirement

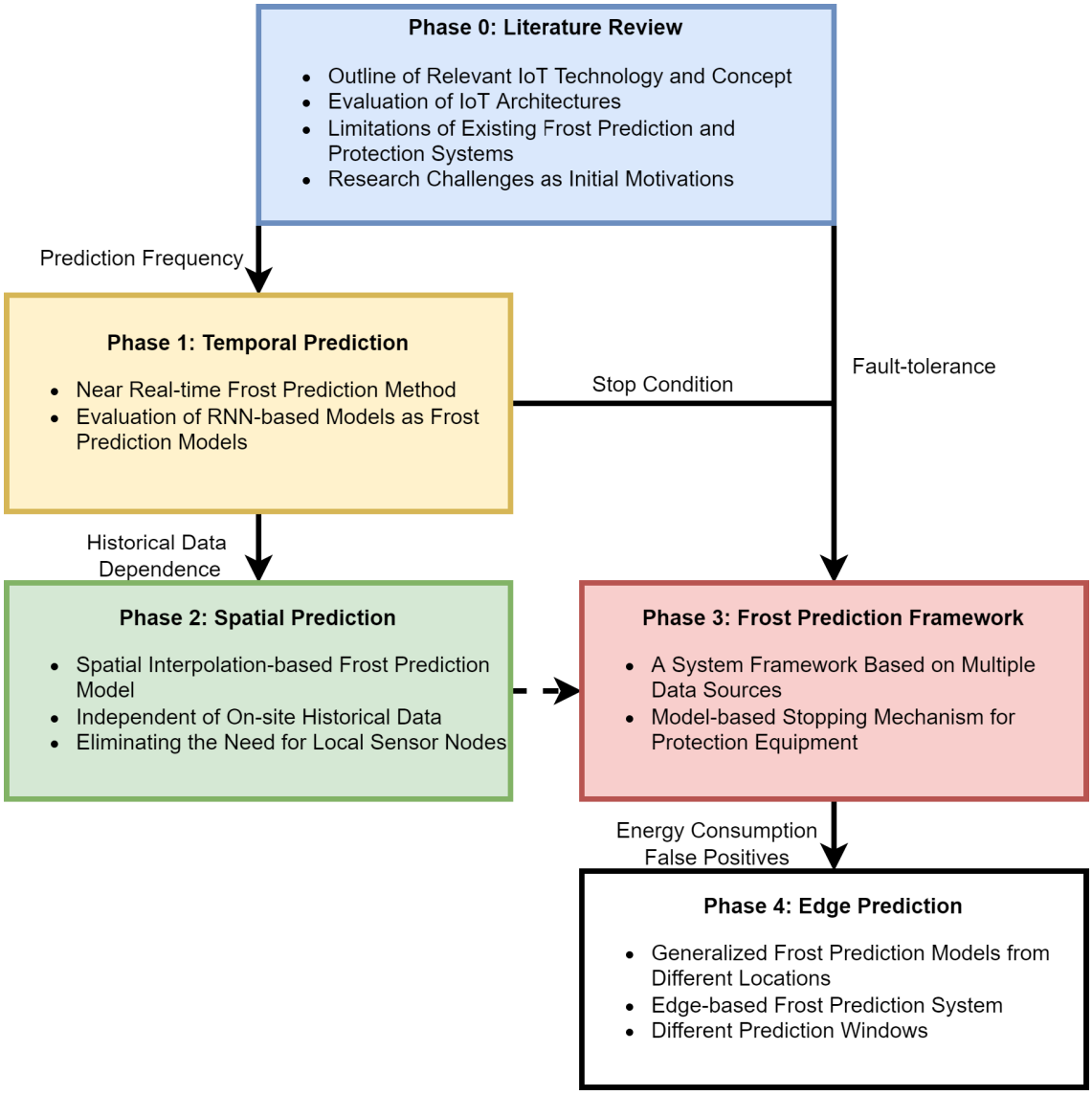


Figure 1.2: Research Methodology.

of local historical data from new sites to construct the models. After that, the results of phases 1 and 2 are merged into the framework in Phase 3. This phase provides a model-based stopping condition for the frost protection equipment. Utilizing external weather stations from phase 2, this framework is able to eliminate the contamination of sensor readings from operating frost protection equipment. Finally, in phase 4, an edge solution of frost prediction is proposed to assess the energy consumption of critical components. Also, the false positives of the system are reduced with lower prediction windows of the frost prediction models.

1.7 Thesis Organization

To complete the objectives and answer the research questions, the rest of this thesis is organized as follows:

- Chapter 2 outlines the relevant technology and concept of IoT relevant to frost prediction and protection under the tree of IoT 2.0. Chapter 2 also identifies the limitations and research challenges of existing frost prediction and protection systems.
- Chapter 3 increases the frost prediction frequency from once per 12–24 hours for the next day or night events to minute-wise predictions for the next hour events. It also evaluates RNN-based models as potential frost prediction models.
- Chapter 4 introduces spatial interpolation methods into frost prediction. This research aims to eliminate the requirement of historical data on models for new sites of deployment and to enhance system fault tolerance.
- Chapter 5 proposes a multi data source frost prediction framework. Model-based stopping mechanism of the frost protection equipment is achieved in this chapter to further enhance system fault tolerance. An implementation of the framework with models from the previous two chapters is also evaluated.
- Chapter 6 presents an edge-based implementation of a frost prediction system. Energy requirements of the edge-based system is compared with a transmission-based system. This implementation also utilizes generalized frost prediction models with smaller prediction windows to reduce the false positives of frost detection.
- Chapter 7 summarizes this research with highlights of each chapter. Finally, challenges for future research are also outlined as the end of this thesis.

2

Literature Review

This chapter provides a comprehensive review of frost prediction systems and introduces its building block technologies under the Internet of Things 2.0. The first part of this chapter presents the concept of the Internet of Things 2.0 with an emphasis on relevant topics to frost prediction and protection. These topics are machine learning intelligence, IoT scalability, IoT interoperability and user friendly IoT. An overview of recent IoT architecture is also provided as reference designs of frost protection systems. Connections between these IoT topics and frost protection are revealed to answer **Research Question 3**: What is the role of IoT in frost protection? The second part of this chapter discusses frost prediction methods, frost protection methods, integrated frost active protection systems, and the IoT communication protocols for frost protection applications. Different prediction methods are compared to answer **Research Question 2**. Moreover, limitations of existing methods and systems are concluded as the response to **Research Question 1** and the starting gaps to be filled in the next sections of this thesis. This chapter is structured as follows. Sections 2.1–2.6 present background information on IoT. Section 2.1 provides an overview of related technologies and concepts. Section 2.2 examines the IoT architectures. Section 2.3 elaborates on the usage of machine learning techniques in different layers of the IoT architecture. Then, Section 2.4 describes different types of scalability and scalability enabled by software defined networks (SDN). Section 2.5 reports IoT interoperability with existing standards. Section 2.6 illustrates user friendly IoT as the final dimension of IoT 2.0. The second part of this chapter (Sections 2.7–2.12) concludes recent development in frost prediction and protection technologies. Section 2.7 discusses the definition of frost. Section 2.8 demonstrates an analysis of current work on frost prediction models. Then, Section 2.9 categorizes the existing frost protection methods into passive and active methods. Then, the limitations of recent automated frost protection systems are

discussed in Section 2.10. Section 2.11 provides insights on the deployment of IoT protocols in terms of power consumption, communication range and cost factors in a frost protection CPS. With the research gaps on frost prediction models and limitations of automated frost protection systems concluded, Section 2.12 proposes future research directions on frost prediction and protection systems. Finally, Section 2.13 offers concluding remarks. Other than these major section, an extra Section 2.14 is included to conclude the development of frost systems after finishing all the chapters in thesis. The major parts of this chapter have been published as the journal papers "*Internet of Things 2.0: Concepts, Applications, and Future Directions*," in *IEEE Access* [33] and "*Frost Monitoring Cyber–physical System: A Survey on Prediction and Active Protection Methods*," in *IEEE Internet of Things Journal* [34].

2.1 Technologies and Concepts underlying IoT 2.0

The term "Internet of Things" (IoT) was first coined by Kevin Ashton in 1999 [35]. The International Telecommunication Union (ITU) has formally defined IoT as "A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies [36]." This definition can be viewed as the basis of IoT technologies. There is an increasing demand for IoT applications and technologies worldwide. It is predicted that networked devices will increase from 18 billion in 2017 to 28.5 billion in 2022, and Machine to machine (M2M) connections will reach 15 billion in 2022 [37]. With recent advancements in the fifth-generation of mobile telecommunications technology (5G), high speed and low latency networks will further facilitate the development of IoT technologies and applications [38]. However, with the recent advancement of other technologies and applications such as machine learning, edge computing, and Industry 4.0, there is a need to update and redefine the concept of IoT towards IoT 2.0 [38–40]. There are many industry and public mentions of IoT 2.0 visions. Many of them focus on improving IoT application productivity and service quality with the vision of users [41–43]. AI-driven service development is viewed as a way to enhance service quality [44]. IoT interoperability is another field that attracted attention for IoT 2.0 [45]. Other than these fields, security and privacy vulnerabilities are also mentioned as issues to be solved in the next generation IoT systems [46]. A potential solution to reinforce IoT security and privacy could be blockchain [47].

At the Samsung Developer Conference 2019, interoperability, security, connectivity, and automation of IoT applications are major fields of further development in the IoT 2.0 vision [48]. Other than this conference, a report [49] from the Joint Research Centre (JRC) of the European Commission concluded that IoT 2.0 should utilize machine learning technologies to enhance the generated intelligence and knowledge available to users. In this process, interpolation is an issue that limits the advancement of specialized edge services. Therefore, approaches toward integration and standardization are inevitable for the evolution of IoT and further development of IoT applications. Compared to the enthusiasm in the industry, academic works on the concept of IoT 2.0 are limited. In [50], an IoT 2.0 platform is proposed. This platform integrates application

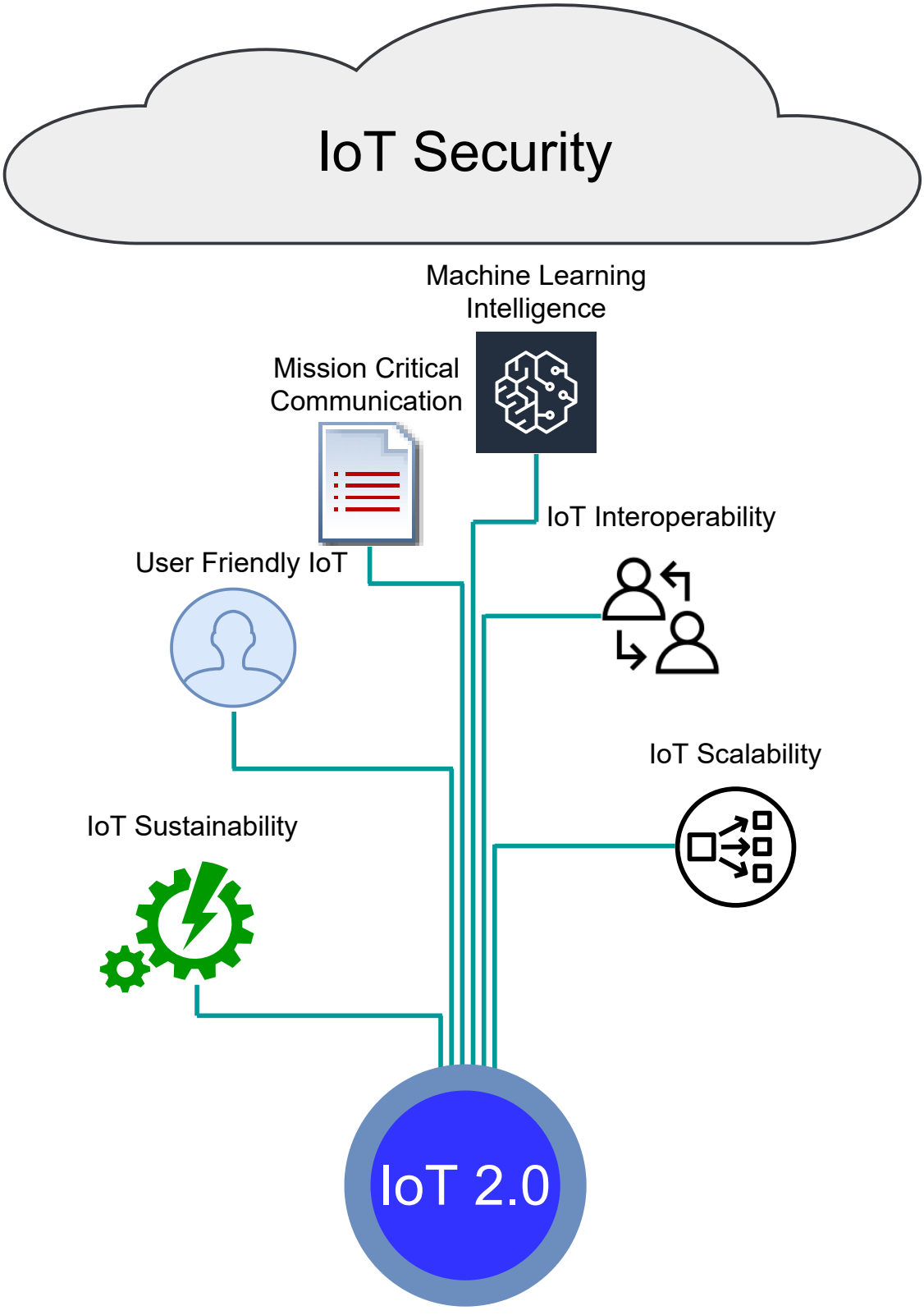


Figure 2.1: IoT 2.0 Concepts.

development, deployment, and sharing. Interoperability is featured as a key function of the IoT 2.0 platform [50, 51]. The authors of [52] demonstrated the "Identity of Things" as an IoT 2.0 component. IoT applications should also be identified by their manufacturers to avoid security issues generated by any criminal parties [52]. In [53], an IoT 2.0 conceptual framework is developed to emphasize the usability of IoT and systems for end-users. Distributed intelligence powered by artificial intelligence (AI) is discussed in [54] and recognized as an aspect of IoT 2.0. The above works only describe one or a few aspects of advancement in IoT. Also, the authors of [55] concluded that very few existing survey papers had connected different aspects of IoT. Recent IoT advancement is concluded into IoT 2.0 as a seven-dimension framework. These dimensions include machine learning intelligence, mission critical communication, IoT scalability, IoT sustainability, IoT interoperability, user friendly IoT, and IoT security shielding the previous six aspects from external attacks (Figure 2.1) [33].The first half of this chapter aims to build up the technical background with the IoT 2.0 dimensions relevant to frost prediction and protection systems.

2.1.1 5G

The authors of [56] revealed the requirements of 5G-based IoT as high data rate, highly scalable and fine-grained networks, very low latency, reliability, resilience, security, long battery lifetime, connection density and mobility. Therefore, 5G grants IoT applications the capability to provide better services by gathering more data in a faster and more secure channel. Furthermore, 5G networks could support the development of next-generation IoT applications. In this subsection, the 5G enabling technologies are reviewed.

Wireless Network Function virtualization (WNFV) is a major part of 5G networks. It not only enables network services to be run through software, but also enables wireless networks to be managed more efficiency and provide better Quality of Service (QoS). Network slicing is key technology within 5G which is built on top of the WNFV to create logically separate networks and provide end-to-end QoS guarantees [57].

5G Heterogeneous networks have evolved to improve the speed of data transmission. To reduce latency, multi-tier cell architectures are deployed to offload data from higher tier centralized cells to lower-tier distributed cells. Lower tier cells are closer to the end users. Hence, latency is reduced [58].

Advanced spectrum sharing and interference management enable wider coverage area and higher traffic load balance [58]. To further improve spectral efficiency, device to device (D2D) communication technology is also included in 5G networks. This technology allows users in close distance to communicate without a base station. Therefore, D2D communication improves not only spectral efficiency but also provides high throughput and energy efficiency [56].

One key enabler of real-time applications is edge computing. As edge computing enables low latency data transmission, real-time smart applications can be developed to provide high quality

services [59]. Therefore, in a 5G network age, integration of AI and edge computing enhanced IoT will significantly enhance the quality of user experience [56].

2.1.2 Tactile Internet

The authors from [60] highlighted that Tactile Internet includes human to machine interactions through haptic devices. They view Tactile Internet on the same level as IoT and 5G. Therefore, revealing the common properties of Tactile Internet, IoT and 5G as low latency, ultra-high availability, Human to Human (H2H)/M2M co-existence, data-centric technologies and security. However, the authors from [61] interprets Tactile Internet as another domain addressed by the low latency requirement of 5G and actuated by the wireless communications of IoT.

Based on the properties of Tactile Internet from [61], the authors of [62] further categorized the challenges of Tactile Internet into communications, haptics, artificial intelligence, and computation. Communication challenges are higher data rates, ultra-low latency, high reliability, and support of cloud/fog network overheads. These requirements are almost identical to the properties of 5G networks. Therefore, communication requirements can be resolved under the environment of 5G. Low latency services also require artificial intelligence and computation power. Artificial intelligence can be leveraged to predict future actions to compensate for latency. Furthermore, artificial intelligence is also the basis of intelligent services. Similar to artificial intelligence, faster computation also reduces the impact of latency. It also supports computation for artificial intelligence and real-time haptic services. The authors of [63] provided six use cases of Tactile Internet applications. The first use case is health care. An example of a health care application is exoskeletons for disabled people. The exoskeleton can detect changes in human muscle to perform certain movements. However, tactile latency is required to ensure safety. Exoskeletons can also be used for education and sports. It can be used in virtual training centers so that students can train in any location. Another use case is traffic. Tactile Internet enables fully autonomous traffic, where vehicles can react instantly to incoming humans on the street. Therefore, this system aims to prevent any injury or death from traffic accidents. This also enhances the performance of monitoring. The usage of free-viewpoint video provides multi-perspective monitoring for users [63]. In the industrial sector, Tactile Internet enables mobile robots for production, reducing any human injuries during production. The last use case is the smart grid. Using Tactile Internet, low latency networks can synchronize the usage of power to the suppliers. This allows the suppliers to precisely adjust the reactive power, preventing wastage of power.

2.1.3 Edge Computing

The aim of edge computing is to reduce bandwidth usage and latency for an IoT network. From Figure 2.2, as a major task of edge computing, machine learning is highly deployable on edge devices [1]. Edge computing is an enabler of low latency and real-time AI applications. In this subsection, the major architectures of edge computing are discussed.

There are three significant architectures of edge computing: fog computing, mobile edge computing (MEC) and cloudlet computing [1]. Fog computing is an extension of traditional cloud computing with fog computing nodes [1]. These fog computing nodes are placed between the end devices and a centralized cloud. The function of these fog computing nodes is to aggregate heterogeneous data from different sources. Furthermore, the fog computing nodes act as an interface to access these heterogeneous data, protecting any user from the heterogeneity of data. In the second architecture, MEC, is designed for cellular networks [1]. Unlike fog computing nodes, MEC nodes utilize both computational and storage capabilities. The functionality of these nodes can be modified through virtualization interfaces. Hence, MEC nodes can provide flexible, low latency, and real-time services to mobile end users. Finally, cloudlet computing is implemented with a cloudlet, which is a virtualized server that is one hop away from the end user [1]. Cloudlets are able to store provisional resources. Therefore, this architecture also can provide end users with flexible, low latency, and real-time services [64]. Based on these major architectures, there are also further enhancements in IoT networks improving energy efficiency [65, 66] and data reliability [67].

In conclusion, the major edge computing architectures are implemented with extended servers or nodes near the end users. The common purposes of these nodes are reducing latency, providing computation or storage capabilities, and delivering real-time services to end users. In a 5G environment, these node properties are the basis of intelligent services pushed by big data transmission and processing. Tactile Internet and Industry 4.0 also drive potential application requirements for IoT 2.0.

2.1.4 Industry 4.0

The authors of [68, p.835] defined Industry 4.0 as "the fourth industry revolution." The Cyber-Physical System (CPS) is the basis of this revolution. The authors of [39] revealed that "CPS are industrial automation systems that integrate innovative functionalities through networking to enable connection of the operations of the physical reality with computing and communication infrastructure." This definition shows that CPSs require heterogeneous data from multiple sources. As a result, data analytics techniques are suitable for implementing intelligence as part of the CPS service. The authors of [39] also pointed out that methods for processing data remain a challenge for these CPS applications. Hence, the implementation of big data analytics and machine learning are essential for the development of Industry 4.0. The amount of data generated by intelligent CPSs is difficult for a centralized cloud architecture to process. Inevitably, edge computing is used to distribute the load for data processing. Also, edge computing devices are closer to the end users. Therefore, it ensures lower latency of a service [69].

2.1.5 Machine Learning

IoT data processing is a challenge due to the 5V (volume, velocity, variety, veracity, and value) features of these data [70]. Data analytics techniques like machine learning can process data with

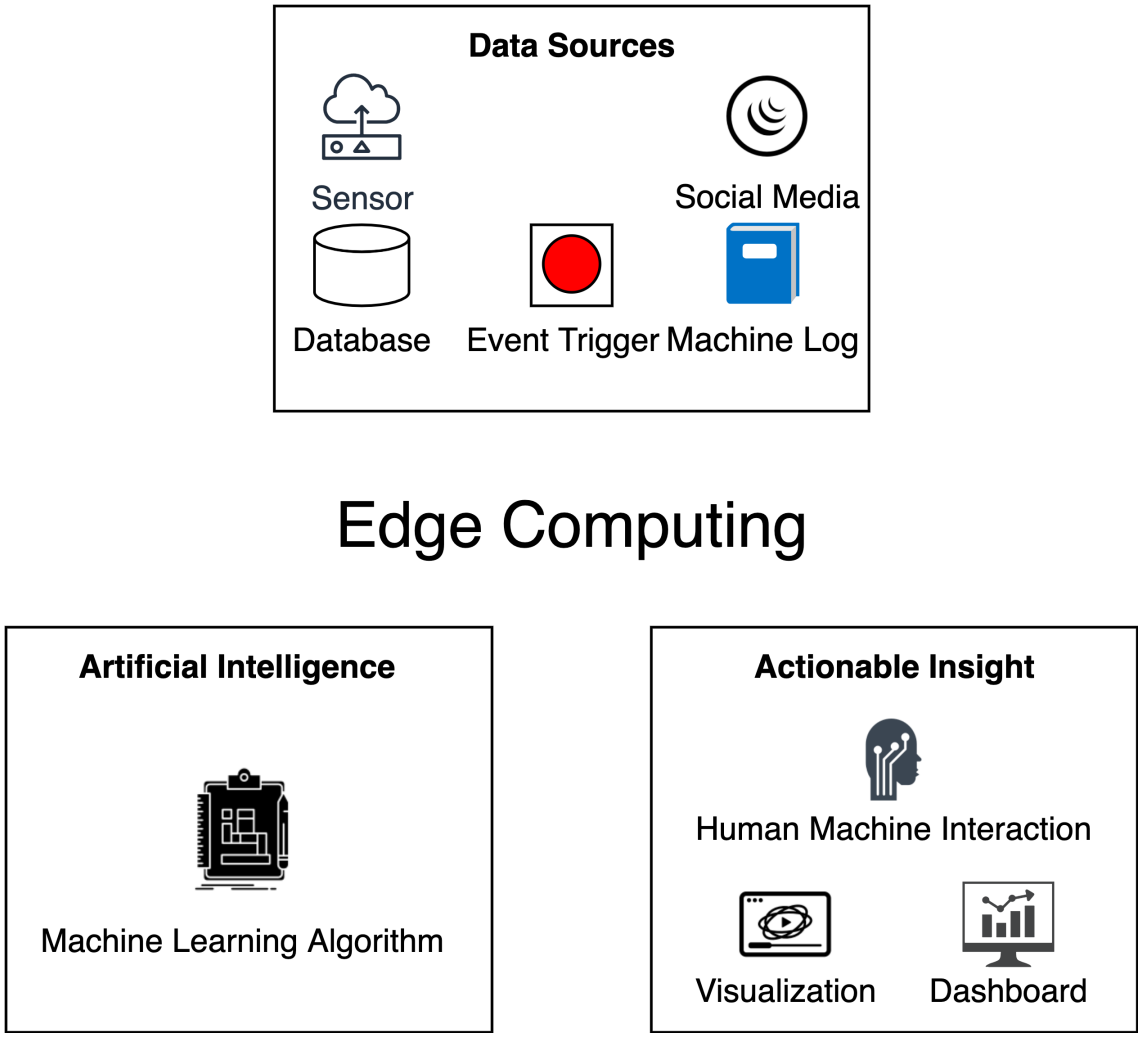


Figure 2.2: Major tasks of edge computing. [1]

complex 5V features [71]. Furthermore, applying machine learning on heterogeneous IoT data ensures intelligence to IoT applications, providing better and efficient services.

The major types of machine learning are supervised learning, unsupervised learning and reinforcement learning [72]. The supervised learning methods use input data with expected outcomes to lead the learning process of a machine learning model. On the other hand, the expected outcome is not provided when training an unsupervised learning model. An unsupervised learning model is built through clustering and other statistical methods [73]. Reinforcement learning models perform actions with input features or state of the current environment. This model learns from the return reward of the action and improves through trial-and-error [4].

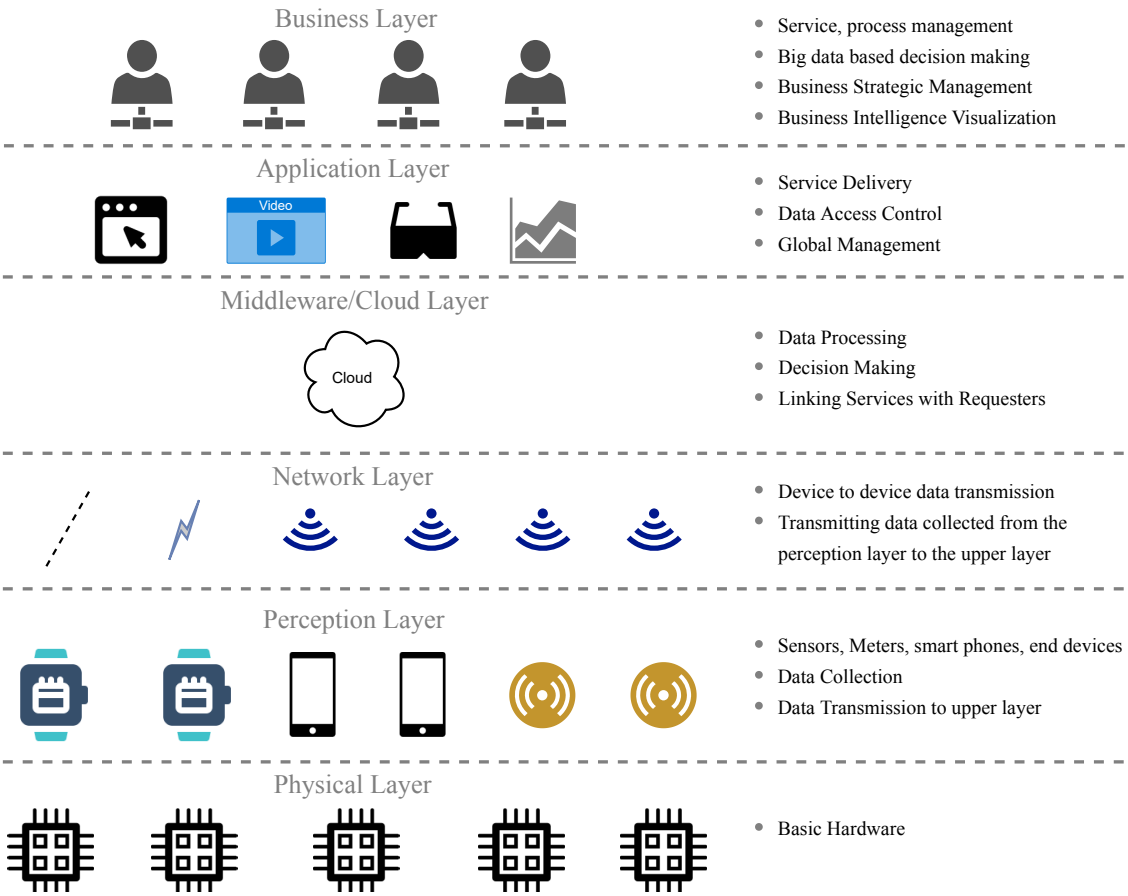


Figure 2.3: Layered Conventional IoT Architecture. [2]

2.2 IoT Architectures

In this section, technical improvements of current IoT architectures are revealed through a detailed analysis of novel IoT architectures under the environment of 5G, Tactile Internet, and Industry 4.0. There are many different IoT architectures [2, 74–81]. The authors in [2] aggregated the conventional IoT architectures into a layered architecture of six layers. From Figure 2.3, this architecture consists of the physical layer, the perception layer, the network layer, the middleware/cloud layer, the application layer, and the business layer. With the assumption that end devices have limited power, memory and computational resource, the perception layer or the end devices in the perception layer are only responsible for data collection and transfer. Therefore, all data is transmitted to the middleware/cloud layer for further processing. For applications with extensive data flows like virtual reality and augmented reality, the throughput and latency of data transmission cannot meet the requirements of real-time, perhaps mission critical service. Therefore, novel IoT architectures are needed in this new era of 5G, Tactile Internet, and Industry 4.0 [74, 82, 83].

Similar to conventional IoT architecture, the recent IoT architectures reviewed in this chapter also contain end-devices and cloud layers. On the other hand, the most significant difference is the utilization of an edge/fog layer in the recent IoT architecture to provide real-time services, data analytics, and data processing functionalities near the end devices. The combination of machine learning models for data analytics services is one of the drivers for these recent architectures [74–81]. Figure 2.4 shows the layers with the functions of these recent IoT architectures. As an architecture providing basic edge computing, the authors followed a three-layered design. This design consists of the IoT end device layer, the fog/edge layer, and the cloud layer. The IoT end device layer is similar to the perception layer of the conventional IoT architecture. This layer also contains IoT sensors, actuators, and end devices for data collection and transmission. Data is passed to the fog/Edge layer to perform analytical procedures and processed for a higher-level layer. The final layer of the three-layered architecture is the cloud layer, providing a platform for centralized data analytics, storage, and decision making [75–77]. Comparing the above recent architectures with conventional IoT architectures, the involvement of the edge computing layer is the root of the changes between architectures.

The authors of [78] separated the cloud layer into a cloud layer and a new network core layer. This layer connects the cloud layer with the fog/edge layer. Also, it provides a flexible and scalable interface for controlling the fog/edge layers [78, 81]. This interface is also developed between the data edge/fog layer and the IoT end device layer. More specifically, the network domain and the communication layer have similar functions to the network layer of the conventional IoT architecture. These layers create a link between the end devices and the fog/edge level devices. Also, as a 5G process, the communication layer facilitates advanced spectrum sharing and interference management for D2D communication [74, 80].

The application layer is above the cloud layer. For different IoT applications, the application layer is different. However, in the recent IoT architectures, the application layer commonly acts as a software interface to control lower layers. Services could be deployed on the cloud level and

Architectures						Functions
					Security	1. Data Encryption. 2. User Authentication. 3. Access Control. 4. Protection over all layers.
					Collaboration/ Process	1. Collaboration platform for multiple personnels
		Cloud Service	Application	Management	Application	1. Software interfaces to interact with lower levels. 2. Services interfaces between lower levels and users. 3. Control interface for lower level facilities. 4. Business Intelligence. 5. High Level Decision Making
Cloud	Cloud	Cloud	Cloud Computing	Server and Storage	Management Service	1. Complex Data Storage. 2. Complex Centralized Processing. 3. Training machine learning models for the cloud and the fog/edge layers. 4. Hosting Complex Machine Learning Model. 5. Basis for High Level Decision Making. 6. Multi-purpose servers. 7. Network management.
	Network Core			Core Network		1. Gateway to the cloud layer. 2. Interfaces to the fog/edge layer. 3. Linkage between the cloud and fog/edge layers.
					Data Storage	1. Storing huge data volume from the edge/fog layer.
Fog/Edge	Fog/Edge	Fog	Data Domain	Edge Network	Edge/Fog Computing	1. Connecting the end devices with the cloud. 2. Gathering data from the lower layer. 3. Basic Data Analytics. 4. Medium Data Analytics (e.g. Image Recognition) 5. Passing heavy analytical tasks to the cloud. 6. Running low level machine learning model. 7. Prepare data as inputs for high level analytics. 8. Data Storage. 9. Low level decision making. 10. Real time services
			Network Domain		Communication	
IoT End Devices	IoT End Devices	Edge Client	Device Domain	End User	Physical Devices	1. Connects lower layer end devices to the upper layer. 2. Management of time sequence sensitive data. 3. D2D Communication. 4. Advanced Spectrum Sharing and Interference Management.
(Peralta et al. 2017; Carnevale et al. 2018; Li, Ota & Dong 2018)	(Shahzadi et al. 2019)	(Kalatzis et al. 2018)	(Chen et al. 2018)	(Guo et al. 2018)	(Rahimi, Zibaeenejad & Safavi 2018)	1. Collecting data from the environment. 2. Transfer data to the upper layer. 3. Accept instructions from the upper layer.

Figure 2.4: Recent IoT Layered Architectures.

the edge/fog level to provide centralized high-level services and distributed, real-time services, respectively [74, 79–81].

The authors of [74] proposed an eight-layer IoT architecture. Different from the previous architectures, the data storage layer, the collaboration/process layer, and security aspects are added to consider the security and performance requirements under the 5G environment. The data storage layer stores raw data from the edge/fog layers. This expands the limited memory of edge devices and prepared for services with high volume traffic. The second layer, collaboration/process layer, is designed for modern business settings. It allows collaboration from different personnel. Finally, security is recognized as a concept applied to all layers to protect them against possible external attacks.

The basic CPS architecture introduced in the following subsection is the basis and generalization of all system architectures proposed in this thesis. This CPS architecture is a reconstruction of the application, cloud, network, perception, and physical (hardware) layers from the conventional IoT architecture (Figure 2.3) highlighting the control aspect. A similar architecture with multiple data sources is adopted in Chapter 5 to increase the fault tolerance of frost prediction systems. The edge computing portion from Figure 2.4 appeared in Chapter 6 as an edge-based frost prediction system.

2.2.1 Cyber-Physical Systems

The fourth industrial revolution or Industry 4.0 leverages CPS as a core element to improve the efficiency and effectiveness of industrial systems [39, 84]. CPSs are hybrid systems of physical and logical elements joined by communication capabilities to provide safe and reliable control on physical entities [85]. Figure 2.5 demonstrates a basic CPS architecture. This architecture consists of sensors for data collection from the environment, actuators to perform an action affecting the environment, intelligence to control the actuators based on sensor data and network to communicate between the prior three elements [3]. In addition to these elements, CPSs also incorporate different technologies such as WSN, IoT and machine learning to achieve the goal of operational efficiency and effectiveness in industrial operations. The rest of this subsection describes the relationship between CPS and these different technologies in the agricultural context.

WSN is a vast network that utilizes many low power and low-cost sensor nodes. With the aid of different communication protocols and network topologies, WSNs can provide manageability to the sensors and actuators within the network [86]. This manageability would aid CPSs to control and operate a high number of sensors and actuators to interact with the environment [87]. In an agricultural context, a terrestrial WSN with reliable and dense communications could be deployed to monitor the environment [88]. The advantages of deploying such a network in agricultural applications are low cost, a low impact on the environment, real-time information feedback, and high efficiency (Table 2.1) [8]. Therefore, WSN is highly compatible with agricultural applications to act as a data gatherer for CPSs.

Table 2.1: Advantages of WSN in Agriculture. [8]

Advantage	**Description**
Low cost	Allows dense deployment to accurately monitor the environment.
Low impact on the environment	Reduces possible cause of stress on crops and animals by the system itself.
Real-time information feedback	Farmers can actively response to environmental changes and adjust their strategy.
High efficiency	Incorporation of sensors and actuators to replace previous manual work.

IoT and CPS are similar concepts. From [85], previous works often separate IoT and CPS from the perspectives of control, platform, internet, and human, but the authors of [85] pointed out that these distinctions are unclear and insufficient. However, CPS is developed from a systems engineering and control perspective, and IoT focuses on networking and information technology [85]. Therefore, in this chapter, the controlling and monitoring ability of CPS is recognized, and the networking and communication ability of IoT highlighted. Thus, IoT is viewed as an autonomous network of wirelessly connected entities through sensors [89]. As CPSs require interaction with different subsystems, IoT can be viewed as an enabler for CPSs to connect with devices for controlling and monitoring [90, 91]. Furthermore, IoT within CPSs is described as Industrial IoT

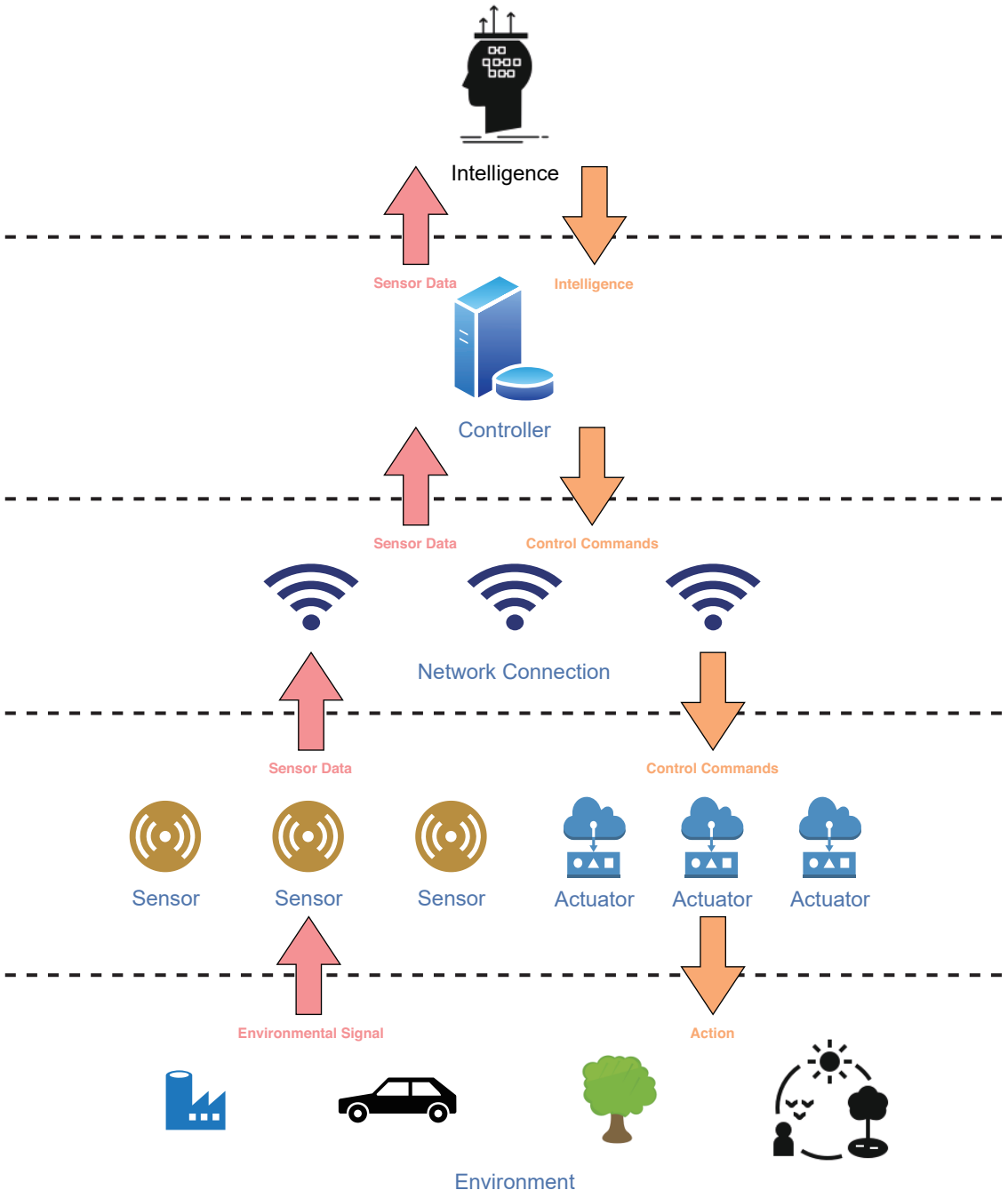


Figure 2.5: Basic CPS Architecture. [3]

(IIoT) [91]. As IIoT provides a massive amount of information, big data analytics are essential to process this information from the sensors [90]. Within the field of big data analytics, machine learning is often incorporated to extract patterns, draw novel insights, and provide intelligence from existing data in IoT networks [71]. Machine learning is also applied to frost prediction with various sensor data.

2.3 Machine Learning Intelligence

This section presents the machine learning intelligence applications. As a start, the relevant supervised, unsupervised, reinforcement, and other relevant machine learning algorithms are introduced. Then, the usage of machine learning on the physical layer, the network layer, the edge computing layer, and the cloud layer are introduced. On the physical layer, machine learning helps end devices perform energy preservation scheduling and physical layer communication. Then, this section demonstrates the usage of machine learning to improve network layer performance and reduce management overhead. After that, edge layer devices and motivations of applying machine learning on edge are described. Finally, this section focuses on the collaboration between the cloud layer and the edge layer.

2.3.1 Machine Learning Algorithms

2.3.1.1 Supervised Learning Algorithms

In supervised learning, the model learns through reducing the output of the cost function, which usually represents the model prediction and the true value. The major supervised learning methods are linear regression, logistic regression, support vector machines (SVM), Naïve Bayes classifiers, and k-nearest neighbors. Some deep learning algorithms, including artificial neural networks (ANN), convolutional neural networks (CNN), and recurrent neural networks (RNN) are suitable for supervised learning [92]. There is a wide range of applications of supervised learning. For example, in the field of computer vision, many CNN-based applications are established in smart healthcare [93], smart home, smart city, smart energy, agriculture, education, industry, government, sports, retail, and IoT infrastructure [94]. The rest of this subsection explains some of the supervised learning algorithms.

Support Vector Machine (SVM) SVM is created to solve binary classification problems [95]. The aim of SVM is to create a hyperplane over a multidimensional space to separate the data points of this space into two classes. The SVM model can be represented by Equation (2.1) [95]. In this equation y is the output class as a sign of positive and negative, ω is the weight vector, x is the input vector and b is the scalar bias factor.

$$y = sign(\omega \cdot x + b) \tag{2.1}$$

From Figure 2.6, the distance between the two classes can be represented by Equation (2.2) [95], where $||\omega||$ is the Euclidean distance.

$$D = \frac{2}{||\omega||} \tag{2.2}$$

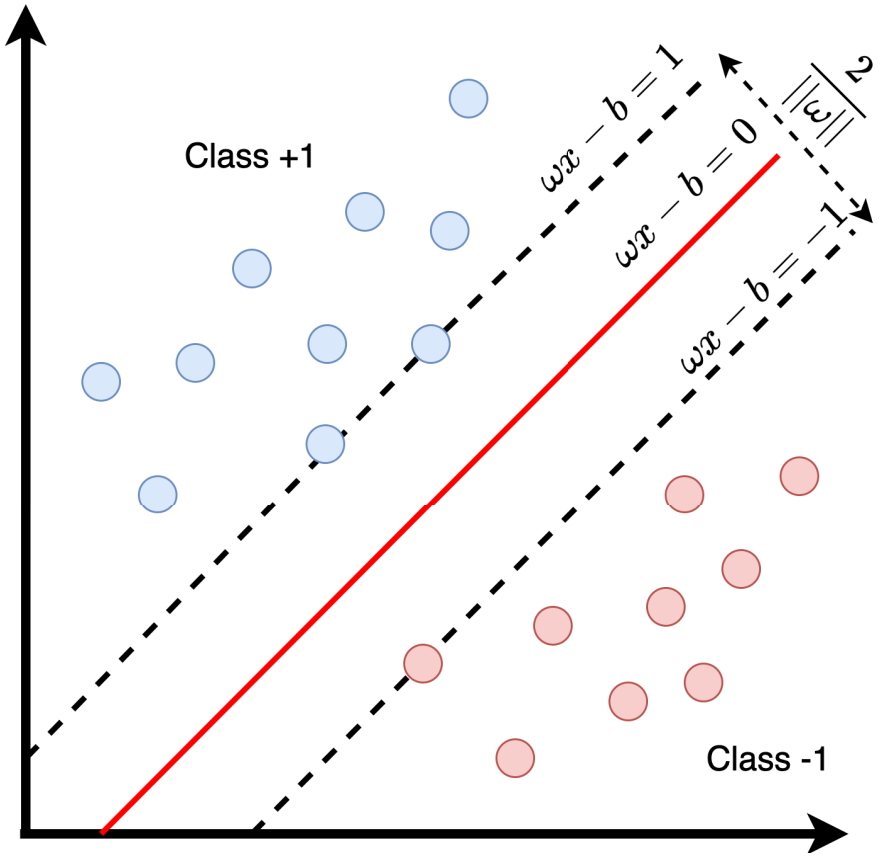


Figure 2.6: Support vector machine.

The parameter ω is obtained through maximizing the distance D with minimum classification error. Therefore, the optimization problem can be defined as Equation (2.3) [95].

$$\Phi(\omega) = \frac{1}{2}||\omega||^2 \tag{2.3}$$

As indicated by [95], optimization of Equation (2.3) is a quadratic optimization problem, which could be solved through constructing a Lagrangian function as Equation (2.4), where α_i are the Lagrange multipliers.

$$L(\omega, b, \alpha) = \frac{1}{2}||\omega||^2 - \sum_{i=1}^{l} \alpha_i\{y_i(\omega \cdot x_i + b) - 1\} \tag{2.4}$$

The SVM described above are only suitable for linearly separable datasets. However, extensions as soft margin SVM and kernel SVM are all capable of handling non-linear datasets. Another form of SVM is the multiclass SVM, which is capable of classifying between more than two classes [95].

Support Vector Regression (SVR) SVM can also be extended to solve regression problems [95]. The generic SVR function is defined by Equation (2.5) [95], where Φ transforms non-linear inputs of x into a higher dimension, the vector w and scalar b should be optimized to minimize the regression risk function defined by Equation (2.6) [95]. In Equation (2.6), C is a constant that represents penalty to errors and Γ represents the cost function. Equation (2.7) [95] defines this cost function with ϵ as the least-modulus loss.

$$f(x) = w \cdot \Phi(x) + b \tag{2.5}$$

$$Rreg(f) = C\sum_{i=0}^{l}\Gamma(f(x_i) - y_i) + \frac{1}{2}||w||^2 \tag{2.6}$$

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \epsilon, & \text{if } |f(x) - y| \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

Finally, similar to the SVM, the optimal parameters can also be found by constructing the Lagrangian function as Equation (2.8) [95]. In this equation, function k is the kernel function to transform inputs into high-dimensional vectors. The variables α_i and α_i^* are the solutions for this optimization problem.

$$\begin{aligned} L = &\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(x_i, x_j) \\ &- \sum_{i=1}^{l}\alpha_i^*(y_i - \epsilon) - \alpha_i(y_i + \epsilon); \\ &\text{Where,} \sum_{i=1}^{l}\alpha_i - \alpha_i^* = 0, \text{AND} \sum_{i=1}^{l}\alpha_i, \alpha_i^* \in [0, C] \end{aligned} \tag{2.8}$$

Linear Regression Linear regression provides an approximation of the relationship between different data domains. In an example of one-dimensional input, the linear regression model is created in the form of the line of best fit (Figure 2.7). The authors in [96] gave a generic model of linear regression with multiple outputs. However, to simplify the process of demonstration, a single-output model is given by Equation (2.9). x and β of Equation (2.9) represent the input vector and the weight vector respectively.

$$f(x) = \beta \cdot x \tag{2.9}$$

The mean squared error (MSE) is computed to be utilized as the loss function (Equation (2.10)). The variable n is the number of data in the training set, x_i represents the ith input vector

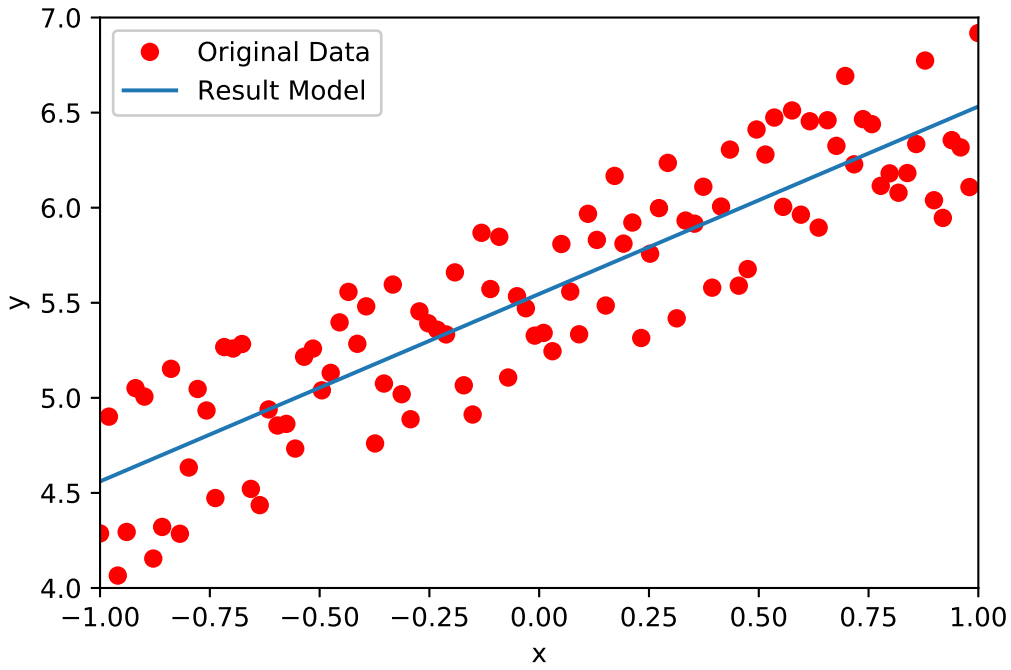


Figure 2.7: One-dimensional input linear regression.

and y_i represents the ith real output.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 \tag{2.10}$$

Logistic Regression The logistic regression solves the binary classification problem. The output of logistic regression is a value between 0 and 1. Thus, providing the confidence level of the prediction. Equation (2.11) demonstrates the logistic regression model, which is based on the Sigmoid function [97]. Similar to the linear regression, β and x are the input vector and the weight vector, respectively.

$$f(x) = \frac{1}{1 + e^{-\beta \cdot x}} \tag{2.11}$$

In order to find the optimal β, the method of maximizing likelihood is leveraged [97]. Equation (2.12) is the loss function. Similar to the linear regression, x_i is the ith input vector and y_i is the ith real output.

$$g = \prod_{i=1}^{n} f(x_i)^{y_i}(1 - f(x_i)^{y_i})^{(1-y_i)} \tag{2.12}$$

However, to ensure this loss function can be processed with an optimization algorithm such as gradient descent, the problem is converted to maximizing the logarithm of the likelihood. This

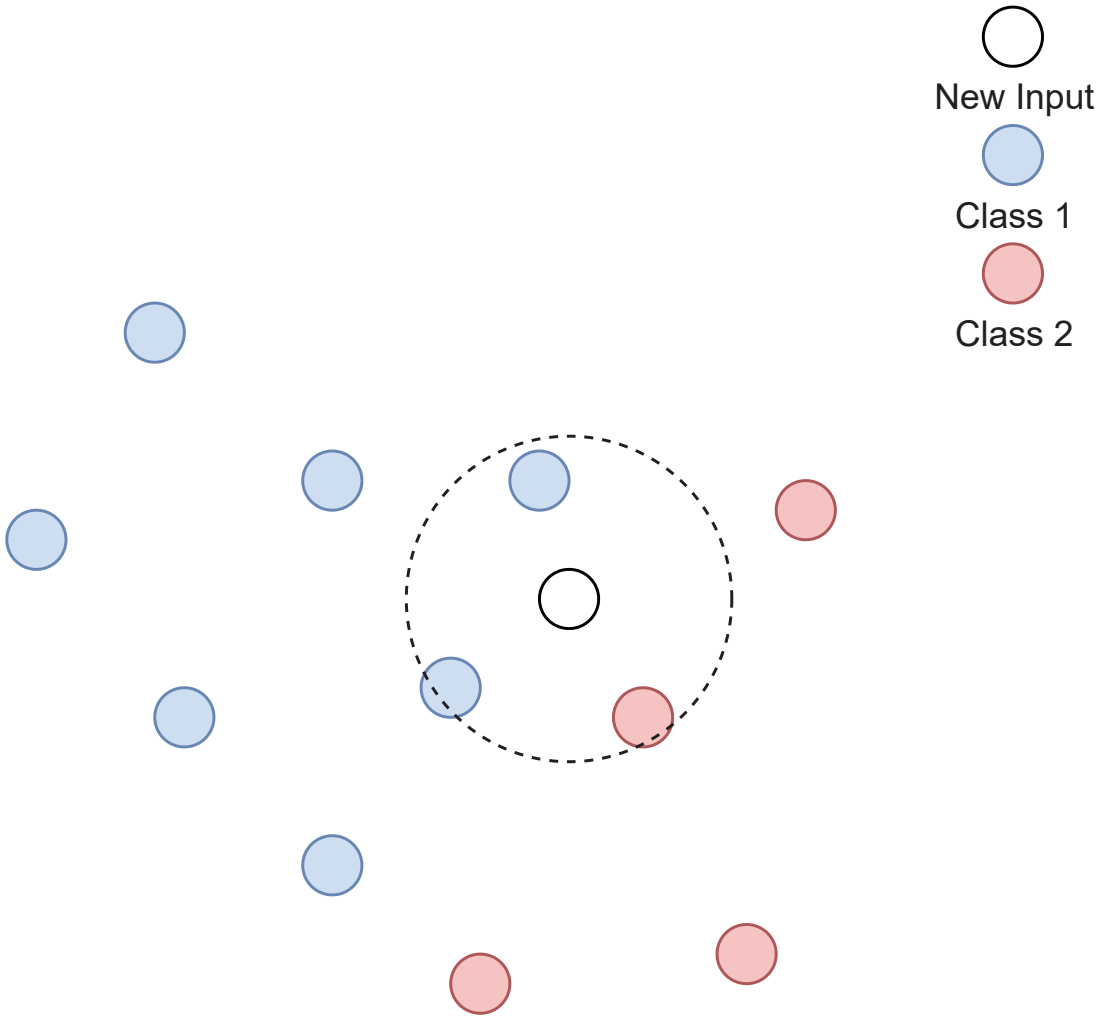


Figure 2.8: KNN inference with three neighbors.

function is presented by Equation (2.13) [97].

$$log(g) = \sum_{i=1}^{n} y_i log(f(x_i)) + (1 - y_i) log(1 - f(x_i)) \tag{2.13}$$

The authors of [97] also provide the general form of the logistic regression using the Softmax function, which incorporates the ability to solve multi-class classification problems.

K-Nearest Neighbor (KNN) KNN is mainly used for classification tasks. The model is built by plotting all training dataset in the feature space. When a new data point is inputted for inference, the model finds K nearest data points in the training set and provides an output based on the majority label of these nearest data points [71]. In order to calculate the distances, distance metrics such as the Euclidean distance, L-infinity norm, angle, Mahalanobis distance, and Hamming distance can be adopted [71]. Figure 2.8 demonstrates KNN with three nearest neighbors. The major label of the neighbors is class 1. Therefore, the new input data point is also labeled as class 1.

Decision Tree (DT) The authors of [98] emphasized that the main objectives of DT classifiers are to limit the classification error to an insignificant level, to classify with high accuracy beyond the training dataset, to achieve incremental updates with new training data, and to structure in a simple form. To achieve the above objectives, algorithms are required to build a DT. Here the ID3 algorithm is used as an example to illustrate DT.

ID3 uses the concept of entropy to construct the DT. Equation (2.14) describes the calculation of entropy, where A is a vector of input features, x_1 and x_2 represent the two classes [99]. Entropy is calculated with all vector A in a tree node.

$$H(a) = \sum_{A}[-P(x_1|A)log_2P(x_1|A) - P(x_2|A)log_2P(x_2|A)] \tag{2.14}$$

New tree nodes should be created with minimal entropy [98]. Therefore, the first step of ID3 is to find an attribute within the input vectors to produce child nodes with the minimal entropy. Then, the input vectors in the root are split according to the attribute to produce the child nodes. Next, if a child node contains input vectors with only one class, the splitting process is terminated for this node and continued with the next child node. On the other hand, if the child node contains input vectors with more than one class, the algorithm repeats the first step with the child node recursively [99].

Ensemble Learning The authors of [100, p.1] defined ensemble learning as "methods that combine multiple inducers to make a decision..." Therefore, as an advantage, models compensate errors of other models. The authors of [100] also divided ensemble methods into the dependent framework and the independent framework. In the dependent framework, the construction of the current model depends on the output of the previous model. An example is the AdaBoost algorithm, where the current model considers the error in the previous model. Gradient boosting machines also adopts a similar concept [100].

The independent framework includes multiple models, which are built independently from each other. Some examples of these methods are bagging, random forest, random subspace methods, error-correcting output codes, rotation forest, and extremely randomized trees [100]. Random forest is described in the next part of this subsection.

Random Forest The random forest is an ensemble learning method based on DT [100]. It consists of multiple DTs. Each DT is trained by a random subset of the training data. Also, another random subset of the attributes is produced for the creation of new child nodes. Therefore, the algorithm only examines part of the attributes for an attribute of the best split. Furthermore, this randomness provides a low correlation between trees, avoiding the domination of a few strong attributes [101].

Naïve Bayes Classifier (NB) NB is a supervised learning algorithm based on the Bayes rule (Equation (2.15)). The Bayes rule provides a model of the conditional probability of a result Y with the given input or the condition X. This algorithm is generally applied to classification problems. In classification problems, Y is from a discrete set of classes. Moreover, an input X belongs to the Y giving the greatest $P(Y|X)$ [102].

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \tag{2.15}$$

The NB model consists of the probability of a class Y and the joint probability of attributes (Equation (2.16)). Therefore, the model is constructed by estimating $P(Y)$ for every class Y in the training set and the conditional probabilities of each attribute $P(X_i = a_i|Y)$ for every class.

$$\begin{gathered} P(Y = y_i|X = a_0, a_1, ..., a_i) = \\ \frac{P(Y = y_i)P(X = a_0, a_1, ..., a_i|Y = y_i)}{P(X = a_0, a_1, ..., a_i)} \end{gathered} \tag{2.16}$$

Bayesian Network (BN) NB models assume that all attributes are independent of applying the Bayes rule. However, in the real world, the correlation between attributes is inevitable [102, 103]. BN is a classifier that is not limited by the assumption of attribute independence. A BN can be represented by Equation (2.17), where G is a directed acyclic graph, where nodes represent the different events and the edges represent the dependency. The symbol Θ contains the Conditional Probability Table (CPT) for all possible values of the attributes and their conditions [103].

$$B =< G, \Theta > \tag{2.17}$$

The learning process is divided into two phases. During the first phase, the graph structure is determined and then in the second phase, the CPT is constructed [104]. The structure can be determined by an expert or learned by data with score-based structure learning methods and constraint-based structure learning methods [105]. The goal of score-based methods is to find a structure that provides the maximum score of a score function. For example, the Bayes Dirichlet equivalent uniform and the Bayesian Information Criterion. In the first step of score-based methods, the algorithm provides a score of suitable parents for every node. Then, parents are assigned to nodes to maximize the scores and to avoid cycles. On the other hand, constraint-based methods use conditional constraints to update the model. An example is the PC algorithm. When using the PC algorithm, the graph starts as a fully connected undirected graph. Edges are removed according to the result of CI tests. This method is repeated until no edges can be removed [106]. After obtaining a graph structure, CPT can be constructed to obtain a full model.

Kernel Bayes Rule (KBR) The KBR extends the Bayes rule by applying kernels to represent probabilities in reproducing kernel Hilbert spaces. Moreover, the prior and likelihood can be expressed by data, which does not require a distribution model [107].

Gaussian Process Regression (GPR) GPR is a non-parametric regression method as the complexity is determined by the data [108]. GPR utilizes the Gaussian Process (GP) to model the function between the input X and output Y. GP is an infinite dimension version of the multivariate Gaussian distributions [108]. GP can be defined by a mean and covariance function. The mean value is usually set as zero and the covariance function can be modeled by a kernel function representing the dependence between different function outputs for different input X [108]. The GPR learning process adjusts hyperparameters of the kernel, such as the length-scale, signal variance, and noise variance [108].

Collaborative Filtering (CF) CF algorithms provide recommendations to a user from experiences of other users [109]. CF operates under two assumptions: Opinions of users do not change over time; Users with similar characteristics provide similar opinions. With these assumptions, CF can be implemented to provide a decision basis for product promotion, social media recommendations, e-commerce reputations, and even strategy [109].

Feedforward Neural Network (FFNN) A sample model of the FFNN or ANN is demonstrated by Figure 2.9. An FFNN contains an input layer, an output layer and one or multiple hidden layers [110].

$$f(X) = f^{output}(f^{hidden2}(f^{hidden1}(X))) \tag{2.18}$$

Equation (2.18) [110] provides the general form of the sample model. In these layers, the input layer consists of the input vector, and the hidden layers can be represented in the form of (2.19) [110], where W is a matrix of coefficients, X is the input vector, B is the bias vector, and g is the activation function. W and B can be learned through the backpropagation algorithm. Whereas, g is chosen by the data analyst to provide nonlinearity [110]. Some candidates of g are the ReLU function, the Sigmoid function, and the Tanh function. Finally, the output layer defines the output type of the model. If the output layer is a Softmax function similar to logistic regression, the FFNN provides the output of discrete values, which solves classification problems. On the other hand, if the output layer provides continuous values like linear regression, the FFNN solves regression problems.

$$f(X) = g(WX + B) \tag{2.19}$$

Convolutional Neural Network (CNN) CNN is a special type of FFNN. CNNs also process input data in a layer-by-layer style. The major motivation of CNN is to reduce the number of parameters to be trained [111]. Figure 2.10 demonstrates the general architecture of CNNs. A full convolutional layer group consists of the convolutional layer, the detector layer, and the pooling layer. In the convolutional layer, the input data is processed by a convolutional filter. This filter is in the form of a vector for one-dimensional data and matrix for two-dimensional data. The filter

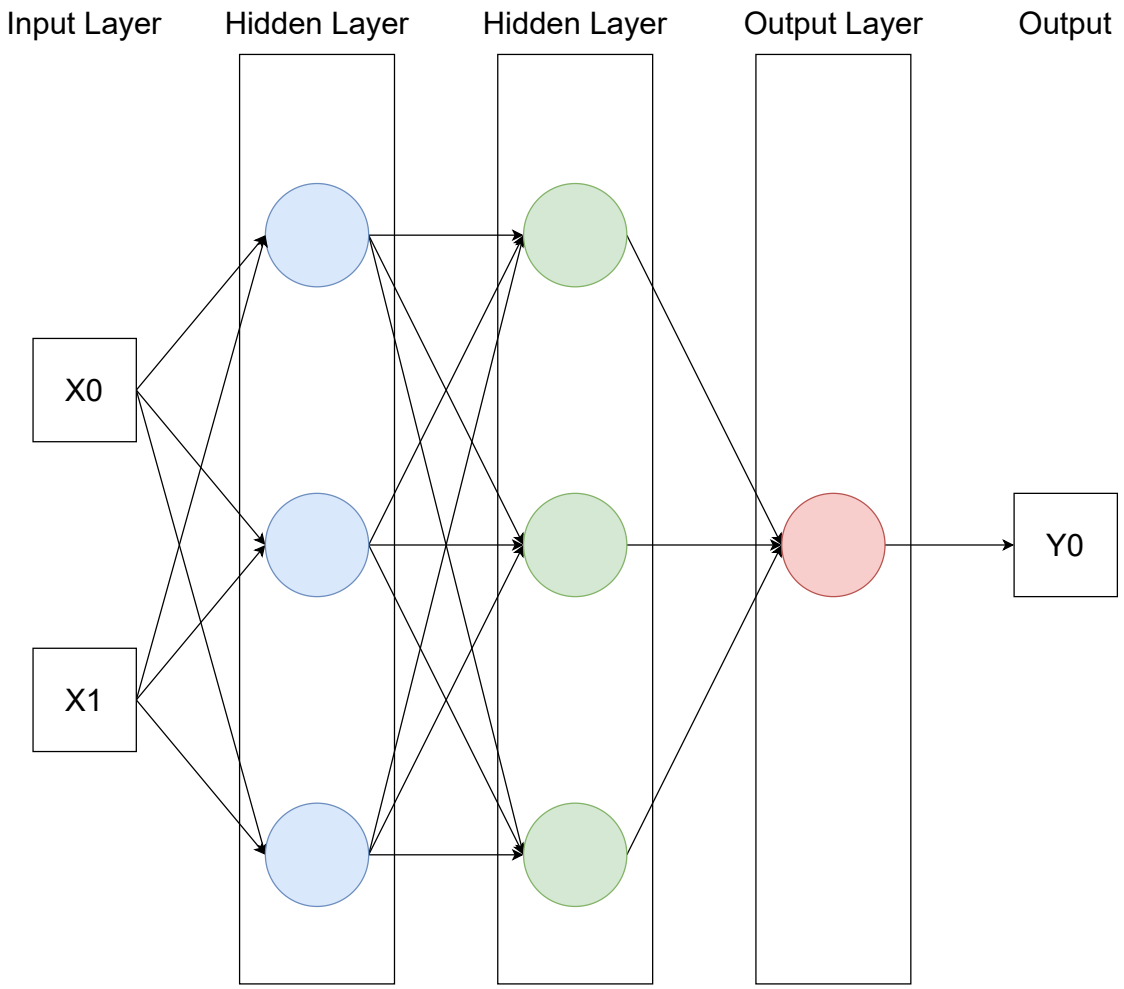


Figure 2.9: Sample Feedforward Neural Network Architecture.

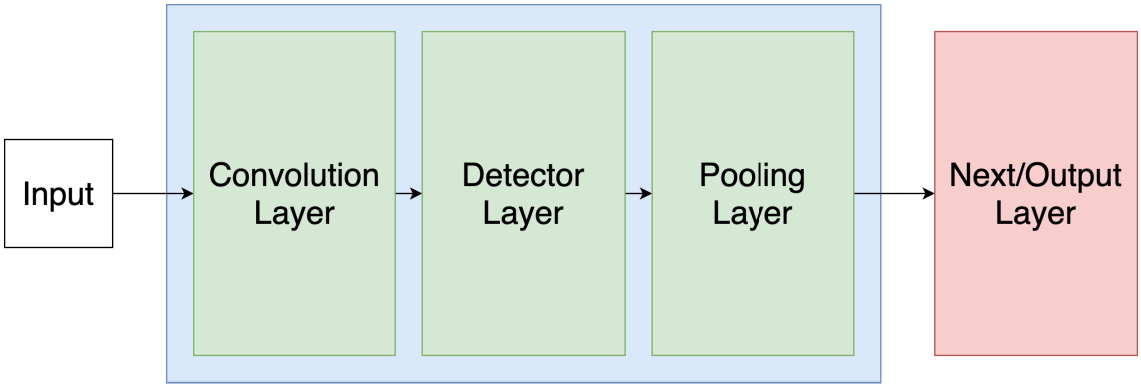


Figure 2.10: General Convolutional Neural Network Architecture.

sweeps through the input data as a moving window, and during each iteration, the dot product of the filter matrix and the current region is calculated. Figure 2.11 provides an example of the first iteration and the final iteration of convolutional layer calculation with 4×4 input and a 2×2 filter.

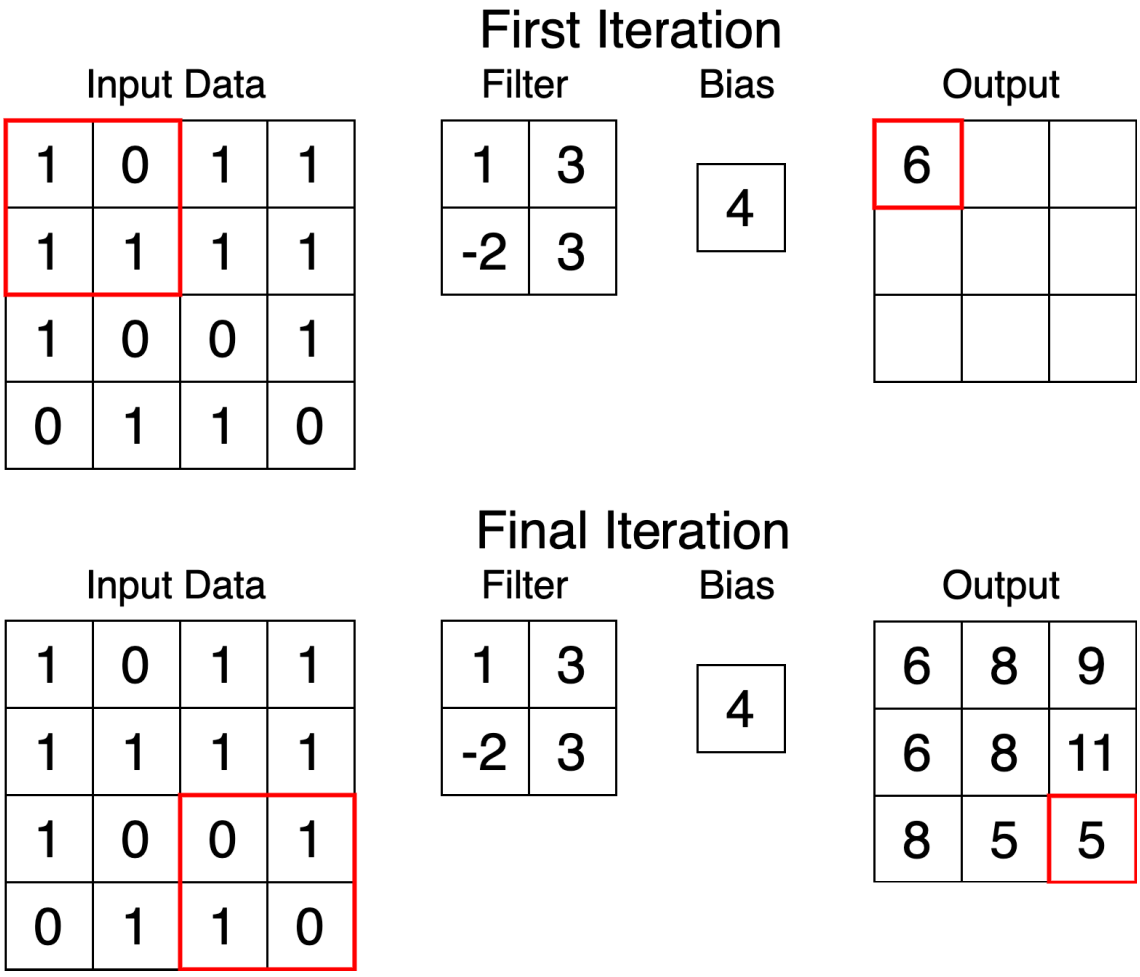


Figure 2.11: 2D Convolution Filtering.

After the convolutional layer, the detector layer processes the data as a hidden layer with the ReLU activation function. The ReLU function provides nonlinearity to the network [110]. Finally, a filter is also used in the pooling layer. Similar to the convolutional layer, the filter in the pooling layer also sweeps through the input. However, the filter only represents the area for the current iteration. Pooling calculation could be simply obtaining the average or the maximum of the filter area [111]. CNN is widely used for image processing.

Recurrent Neural Network (RNN) Unlike the basic FFNN, which only accepts one input a time, RNNs accept several inputs [112]. In terms of time-series data, individual data points are processed at once in the sequence of time [112]. As shown in Figure 2.12, the output of the current hidden state H^t is generated from the input X^t of the current time state and the output X^{t-1} of the previous time state, recursively [112]. Finally, if only one output is required (for classification or regression), the output Y is calculated from the final hidden state [112].

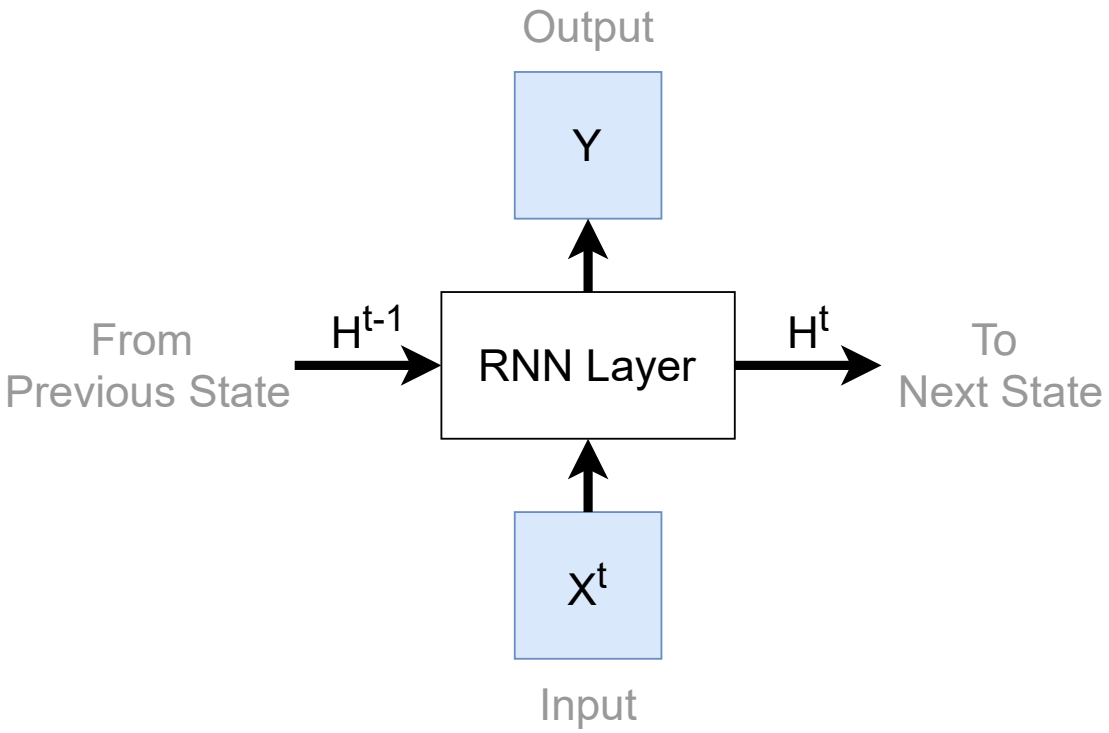


Figure 2.12: RNN Layer at time t.

Long Short-term Memory (LSTM) Since gradient propagates through multiple stages in RNNs, issues such as gradient explosion and gradient vanishing arise [112]. To address these issues, Long Short-Term Memory (LSTM) is proposed as variants of the RNN [112]. The LSTM incorporated an additional cell state to enhance long term memory [113]. Also, the additional forget and input gates are utilized to forget and insert information into the cell state [113].

Random Neural Network (RandNN) The RandNN is a type of RNN. Excitatory impulse signals of "*+1*" and inhibitory impulse signals of "*-1*" are transmitted between the neurons of RandNN [114]. The neuron state or potential at a certain time is represented by a non-negative integer. This potential increases when the neuron receives an excitatory impulse and decreases when the neuron emits a signal. The neuron emits signals when its potential is positive. Also, the acceptance of an inhibitory signal outside of the network decrements the neuron potential [114]. The RandNN can be applied in multiple fields such as associative memory, optimization, texture generation, magnetic resonance imaging, function approximation, mine detection, and automatic target recognition [114].

2.3.1.2 Unsupervised Learning Algorithms

The two major types of unsupervised learning models are principal component analysis (PCA) and K-means clustering. PCA is used as a technique to compress data. This is important for IoT

applications, such as wireless sensor networks (WSN), with limited throughput and energy [115]. The K-means algorithm is used for the clustering of multiple sensors. By dividing the monitored field into areas using the unsupervised K-means clustering, the complexity of data gathering and processing are reduced [115]. Some other unsupervised learning algorithms are also explained further in this subsection.

K-means The K-means algorithm produces a classification model through clustering [116]. It aims to generate multiple K centroids from the dataset. Data points close to a centroid forms a cluster [116]. The centroids are initialized by choosing random data points from the dataset. Then, data points are assigned to the cluster of the nearest centroid. Next, the new K centroids are calculated by averaging the assigned data points within their clusters. The above steps are iterated until the centroids are stable, or the algorithm reached a preset number of iterations [116]. With the centroids calculated, a data point can be classified by computing the distance towards the centroids. The new data point belongs to the cluster of the closest centroid [116].

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) DBSCAN is another clustering method similar to K-means. However, compared to K-means, DBSCAN does not require a predefined number of K centroids. Also, DBSCAN can identify noises. Moreover, the shape of the cluster can be arbitrary [117]. DBSCAN has two hyperparameters the minimum number of neighbor points $minPoints$ within the distance R [117]. To construct the clusters, DBSCAN iterates through all points in the dataset [117]. If an unvisited data point has more than $minPoints$ neighbors within R, the data point is marked as a core point, and a new cluster is created. After that, recursively, all previously unvisited neighbors of the core point are visited and added into the cluster. Also, if the neighbor point is another core point, the two clusters would merge [117]. If a data point has less than $minPoints$ of neighbors within the range R, the data point is classified as noise [117].

Hierarchical Clustering Analysis (HCA) HCA is a clustering method, where the data sample is recursively merged or split to form a tree diagram [118]. HCA methods can be divided into agglomerative hierarchical clustering and divisive hierarchical clustering. Agglomerative hierarchical clustering is the bottom-up approach, where each data point forms its own cluster, and similar clusters merge until the desired architecture is obtained. On the other hand, divisive hierarchical clustering is the top-down technique as it starts with a huge cluster containing the whole data sample. Then, the cluster is divided to form the tree [118]. Merging and division decisions are made with similarity criteria. The three different sets of criteria are single-link clustering, complete-link clustering, and average-link clustering. For the three clustering methods, the distance between two clusters is calculated as the shortest distance between any two members from different clusters, the longest distance between any two members from different clusters, and the average distance between any two members from different clusters, respectively [118].

Expectation Maximization (EM) The EM algorithm computes maximum likelihood estimations for latent variables [119]. The algorithm consists of the Expectation (E) and Maximization (M) steps. The E step computes the missing data from current function parameters. During the M step, the function parameters are updated to maximize the log-likelihood of the estimated latent variables [119]. The E and M steps are repeated until the model converges slowly to a local maximum [119].

Gaussian Mixture Modeling (GMM) The superposition of multiple Gaussian distributions can approximate any continuous density through the adjustment of their means, covariances, and coefficients [120]. Unlike the parameters of a single Gaussian model that can be determined directly by the maximum likelihood method, GMM is trained using EM in an iterative way [120]. GMM can be applied to solve clustering problems [120].

Principal Component Analysis (PCA) PCA reduces the number of attributes in a dataset by transforming the original inputs into another set of vectors with low information loss [121]. Dimensionality reduction is achieved by eliminating components with a lower variance. These components are detected through the computation of the eigenvectors and eigenvalues of a covariance matrix from the original dataset [121]. A component with a higher eigenvalue indicates more information contained. Therefore, features can be extracted by choosing the corresponding components or eigenvectors with higher eigenvalues [122].

MultiDimensional Scaling (MDS) MDS is another dimensionality reduction technique. However, unlike PCA, MDS preserves the distance or difference between sample cases instead of the variance [123]. Stress, the loss function of MDS is defined as Equation (2.20), where d_{ij} is the difference between sample cases i and j in the original data space, and D_{ij} is the distance between i and j in the lower dimension or projected data space [124]. MDS consists of four steps [125]. In the first step, a squared distance matrix is computed from the data points in the original data space. Then, the matrix B is computed by applying double-centering to the squared distance matrix. After that, the eigenvalues V and eigenvectors Q of matrix B can be obtained. V_m is a matrix of the first m eigenvalues greater than zero, and Q_m is a matrix of corresponding eigenvectors. Finally, the coordinate matrix can be calculated by multiplying Q_m and $V_m^{\frac{1}{2}}$ [125].

$$Stress = \frac{\sum_{i=1,j=1}(d_{ij} - D_{ij})^2}{\sum_{i=1,j=1} D_{ij}^2} \tag{2.20}$$

Diffusion Maps (DM) DM is also an algorithm for dimensionality reduction [126]. In contrast to PCA and MDS, DM unravels the potential manifold structures in the dataset [126]. The DM algorithms initiate by defining a kernel and a kernel matrix. Through normalization of the kernel matrix, the diffusion matrix can be acquired. Finally, DM utilizes n numbers of the most dominant

eigenvectors from the diffusion matrix to project the dataset from the original data space to n-dimensional diffusion space [127].

Window Sliding with De-Duplication (WSDD) WSDD is used to mine patterns from system events sorted in chronological order [128]. WSDD utilizes a sliding window over the training dataset to learn patterns in a brute force approach. The algorithm is capable of detecting both frequent sequential patterns and periodic sequential patterns [128]. To increase efficiency, the algorithm stores mined patterns in a hashmap and avoided mining duplicate patterns. The pattern itself is stored as the key in the hashmap, and the count of the pattern is stored as the value. Finally, only patterns detected over a minimum count are returned as the output of WSDD [128].

Autoencoders (AE) The AE is a neural network consisting of the encoder, code, and decoder components [129]. The encoder maps the raw input to the output of the code component, and the decoder reconstructs the raw input from the output of the code component. AEs can be used for feature reduction as the output of the code component from a trained AE holds near lossless information of the raw input [129].

Hopfield Neural Network (HNN) The HNN is a type of RNN for solving optimization problems [130]. Each neuron provides non-linear outputs through a sigmoid function. All neurons are interconnected with each other to restrict and revise the outputs of each other. Each connection includes an interconnection weight. Each neuron contains a user adjustable input bias [130]. The neurons update according to the energy function (Equation (2.21)), where T_ij is the weight of the connection between neurons i and j, V is the output of a neuron [131]. The HNN neurons evolve until a local minimum of the energy function is reached [131].

$$E = -\frac{1}{2}\sum_{i \neq j} T_{ij} V_i V_j \tag{2.21}$$

Self-Organizing Map (SOM) The SOM is a type of neural network that can perform clustering similar to the K-means [116]. In each iteration, the neuron closest to a randomly selected data point moves towards the data point by a preset learning rate [132]. Neurons within the preset neighbor range of the first neuron also move towards the data point. The learning rate and the neighboring radius delays over the number of iterations [132].

2.3.1.3 Reinforcement Learning Algorithms

The goal of reinforcement learning is to solve the problem of Markov decision processes (MDP). MDP is a sequential decision problem. As demonstrated in Figure 2.13, any action made by the agent will influence the environment and generate a reward. The goal of reinforcement learning is to maximize long-term rewards [4]. Q-learning, a type of reinforcement learning, is used to solve

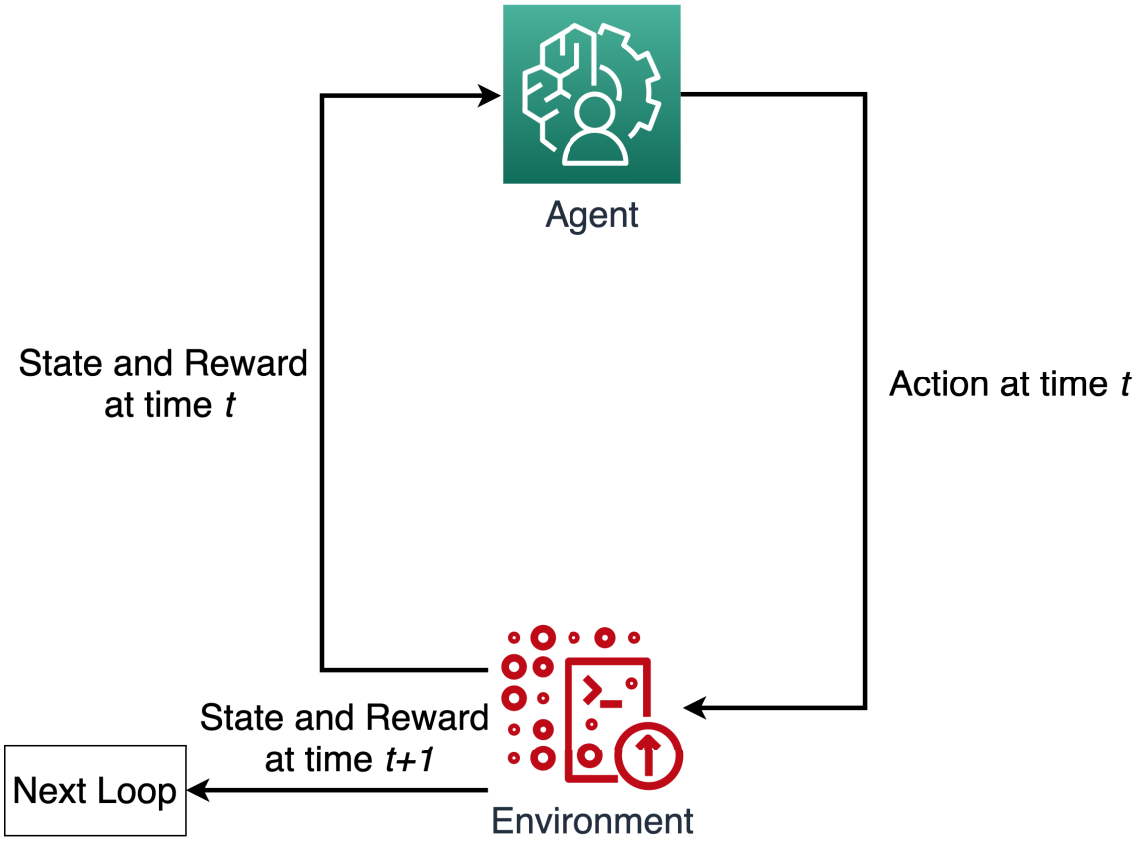


Figure 2.13: Agent-Environment Relationship. [4]

routing problems in IoT networks. Unfortunately, most of these algorithms are based on wired networks [133]. In WSNs, energy, processing power, and storage might become a bottleneck for distributed reinforcement learning routing algorithms [70]. Reinforcement learning algorithms aim to provide high-level intelligence to IoT applications.

Temporal-Difference (TD) TD learning includes various model-free reinforcement learning algorithms, which require no model of the environment [134]. TD algorithms bootstrap or update the estimates based on current estimations. The value function is updated at every step of TD [134]. There are three fundamental types of TD-based learning algorithms mentioned in the sections after. The on-policy TD algorithm **SARSA** learns the action values from the current policy, while the off-policy algorithm **Q-learning** learns from the optimal policy [134]. Finally, a third type of TD learning, the **Actor-critic learning** learns both a policy (actor) and value function (critic) [135]. Actor-critic learning is always on-policy as the "critic" needs to learn from and correct the TD errors from the "actor" or the policy.

Least-Squares Policy Iteration (LSPI) LSPI is a model-free off-policy reinforcement algorithm [136]. LSPI is also an approximate policy-iteration algorithm, where the value function and policy representation are approximated. Therefore, compared to tabular methods, the policy

search process is more efficient for LSPI [136]. Also, LSPI is based on least squares temporal difference learning [136]. Thus, as TD learning methods update incrementally, data efficiency of LSPI can be preserved [134].

2.3.1.4 Other Relevant Machine Learning Techniques

Transfer Learning By adopting transfer learning techniques, a model trained to solve one problem can be transferred and adapted to solve a different problem [137]. This prevents time-consuming labeling processes. Transfer learning can be categorized into inductive transfer learning, transductive transfer learning, and unsupervised transfer learning [137]. In the inductive transfer learning setting, the domains can be the same or different, but the tasks are different for the two problems. Whereas in transductive transfer learning, the tasks are the same, and the domains are different. Finally, in unsupervised transfer learning, similar to inductive transfer learning, the tasks are different. However, unsupervised transfer learning performs unsupervised learning tasks in the target problem [137].

Federated Learning Federated learning is a technique of multiple users training a common machine learning model without leaking their local dataset to other users [138]. There is the horizontal federated learning technique, where different datasets share the same features, but different sample cases [138]. On the other hand, vertical federated learning can be applied to datasets with more overlapping sample cases and different features [138]. Finally, federated transfer learning is suitable for datasets with different sample cases and features [138].

2.3.2 Physical Layer Applications

One major application of machine learning influencing IoT end devices is communication control. The authors in [139] used Q-learning for transmission power control to reduce the unnecessary waste of power due to interference. This model is only tested under the scenario of one-to-one transmission. A scenario of multiple sources toward multiple receivers should be tested.

The authors of [140] explored the usages of deep learning in end-to-end communication systems. The authors adopted the AE to replace different compensation techniques during the transmission of data. Data is encoded between transmission and decoded after transmission to protect the payload during transmission. Another application is the implementation of CNN for modulation classification. This is a prerequisite for developing an intelligent receiver.

Machine learning algorithms increase the energy consumption of IoT devices. Therefore, it is important to apply energy preservation techniques. The authors of [141] concluded that the two major energy preservation methods are energy saving and energy harvesting. Most of the energy saving techniques are implemented through the estimate and control of the uptime of end devices [139, 142–147]. The rest of this subsection focuses on machine learning-based energy saving techniques. The authors of [142] established ARIIMA or A Real IoT Implementation of

a Machine-Learning Architecture for reducing energy consumption. This is an IoT architecture that uses machine learning to forecast end device usage to control the up and downtime of IoT end devices. The aim is to reduce energy consumption. The authors compared different methods of calculating the loss of the predicted outcome. However, the authors did not link energy efficiency improvement to any specific machine learning algorithms.

The authors of [144] utilized the Naïve Bayes Classifier for calculating the optimized uplink period for IoT data collectors. The goal of this work is to optimize the uplink time for power efficiency and preserve the accuracy of data.

The authors from [146] used a single hidden layer feedforward neural network to predict the power usage based on smart meters. With these predictions, the power suppliers can balance the power production with consumption to avoid power wastages. Also, individual home devices can be controlled to relieve the grid pressure at power peaks.

The authors of [145] used logistic regression, KNN, and Naïve Bayes algorithm to increase the power efficiency of smart buildings. Light, temperature, and motion data of a room are fed into the models to determine whether if people are present in a room. In conclusion, this work only determines the existence of people. Nevertheless, further work needs to be done on the development of an energy efficient device control scheme based on the predictions of these machine learning models.

The authors of [147] extended the model for predicting human presence in smart buildings. A random neural network model is applied with inputs of carbon dioxide level and temperature readings to predict the number of occupants in a room. This model is used to control the heating, ventilation, and air conditioning (HVAC) systems. HVAC devices will be turned off to save power if no occupants are detected in the room.

The authors from [143] pointed out that the manual labeling of training data is time consuming in supervised learning algorithms. Therefore, the authors proposed an energy saving scheme based on unsupervised learning. The WSDD algorithm is used to extract patterns of device behavior from historical data.

2.3.3 Network Layer Applications

The authors of [9, 10] summarized existing network layer applications using machine learning algorithms. These applications are IoT device identification, network routing, traffic profiling, traffic prediction, traffic classification, congestion control, resource management, fault management, QoS and Quality of Experience (QoE) management, and network security. Table 2.2 links these applications to implemented machine learning algorithms. However, these applications alone might not be feasible to deal with the complexity of networks such as 5G, Tactile Internet, and Industry 4.0 requirements. Furthermore, an autonomous network structure is required.

Table 2.2: Network Applications and Related Machine Learning Algorithms. [9, 10]

Applications	**Machine Learning Algorithms**
IoT Device Identification	KNN, SVM, GMM, decision tree, ensemble learning, random forest
Network Routing	LSPI, Q-Learning, n-step TD, SARSA
Traffic Profiling	K-means, Clustering
Traffic Prediction	FFNN, SVR, KBR, LSTM, GPR
Traffic Classification	SVM, NB, HCA, KNN, DT, K-means, Random Forest, FFNN, DBSCAN
Congestion Control	EM, DT, Random forest, KNN, FFNN
Resource Management	FFNN, RandNN, SVM, HNN, RNN, Q-Learning, TD, BN
Fault Management	BN, FFNN, DT, SVM, Ensemble Learning, Linear Regression, Autoencoders, K-means, EM, RNN, SOM
QoS and QoE Management	FFNN, DT, Random Forest, NB, SVM, KNN, SVR, Q-learning
Network Security	FFNN, Ensemble Learning, DT, BN, NB, SVM, KNN, Linear Regression

2.3.3.1 Self-organizing networks

The increasing network complexity and device numbers for 5G and beyond networks are inducing conflicting demand over network resources and routing decisions. Therefore, self-organizing networks (SON) are required to reduce the complexity of managing these networks [148]. Management functionality of SONs consists of self-configuration, self-optimization, and self-healing. Self-configuration processes automate network design, network planning, and network deployment. After that, the self-optimization functionalities maintain the network performance and conduct routine network operations [149]. Finally, self-healing functionalities focus on fault detection and recovery [150].

The authors of [151] organized machine learning in SONs into four modules: sensing, mining, prediction, and reasoning. Sensing involves the detection of network anomalies and routine events. Therefore it contains functionalities of self-optimization and self-healing. Mining aims to classify services to help the network to optimize its performance. Moreover, mining belongs to the self-configuration functionalities. Finally, reasoning could apply to the offline parameter tuning during self-configuration and the online parameter tuning for self-optimization during network runtime.

The authors of [11] categorized machine learning applications on SONs according to the three functionalities. In Table 2.3, the self-configuration applications are operational parameters configuration, neighbor cell list configuration and radio parameters configuration. In Table 2.4, the self-optimization applications consist of backhaul, caching, coverage and capacity, mobility, handover, load balancing, resource optimization, and coordination. In Table 2.5, the self-healing applications include fault detection, fault classification and outage management. Table 2.3, 2.4 and

Table 2.3: Machine Learning Applications in Self-configuration. [11]

Applications	**Description**	**Machine Learning Algorithm**
Operational Parameters Configuration	Configuration of the base station for basic operations.	SOM
Neighbor Cell List Configuration	Neighbor discovery, Self-advertisement	N/A (Control-based algorithms)
Radio Parameters Configuration	Transmission power, radio angle, topology configuration.	Q-Learning

2.5 only include the algorithms that are relevant to supervised learning, unsupervised learning and reinforcement learning. Therefore, controller models, Markov models, and heuristics algorithms are out of the scope of this chapter.

The authors from [152] promoted self coordination as a fourth functionality group of SONs. Their work demonstrates that the current design of standalone management functionalities of SONs could lead to conflicting parameter adjustment between distinct functions. This work also concludes that DT, Q-learning, actor-critic learning, and SVM can be solutions for self-coordination.

The authors from [153] proposed another method to avoid collision between different functionality results. Their distributed Q-learning model considers both base station power allocation and user quality of service. Q-learning consists of a list of actions, a list of states, and a list of rewards. The actions are the power allocation for the base stations. The states are the ring that the agent is covered with current power allocation. Finally, the rewards are calculated considering the higher capacity of the base station and better user quality of service.

The network applications for traditional networks in Table 2.2 could be applied to support the SON functionalities. The authors of [154] emphasized that the result of traffic forecasting and prediction can increase the performance and accuracy of all other SON functionalities. The authors tested three types of machine learning models for traffic forecasting. The first type of model is autoregressive algorithms. This includes linear or polynomial regression. The second type of model is neural networks and finally, the authors used GPR for traffic forecasting. The authors also mentioned that this application can be further extended for QoS management and congestion control, providing possible use cases for models in the traditional networks. To improve the current management scheme in 5G and beyond networks, the implementation of SDN and Network Function Virtualization (NFV) architectures in SONs fulfills the intelligence, automation, management, and optimization requirements [155]. In this architecture, machine learning works at the core to enable intelligent network management. This work also demonstrates that traffic classification as an essential application provides results affecting consecutive decision making processes.

Table 2.4: Machine Learning Applications in Self-optimization. [11]

Applications	**Description**	**Machine Learning Algorithm**
Backhaul	Connection between user, base station and the core network.	Q-Learning
Caching	Temporarily storing functions and services on the base stations	CF, K-means, Game Theory, Q-learning, Transfer Learning
Coverage and Capacity	Managing tradeoff between network coverage and network capacity	SOM, Q-learning
Mobility	Locate and predict the location of the user.	Naïve Bayes classifier, FFNN, SVM, DT, K-means
Handover	Realtime change of channel parameters when the user is using the channel. Often associated with mobility management when users move between cells.	FFNN, SOM, Game Theory, Q-learning,
Load Balancing	Intelligently balancing network load	Q-learning
Resource optimization	Allocation and prediction of network resource usage.	FFNN, K-means, SOM, Game Theory, Q-learning, Transfer Learning
Coordination	Minimizing the interference between two different functionalities.	DT

Table 2.5: Machine Learning Applications in Self-healing. [11]

Applications	**Description**	**Machine Learning Algorithm**
Fault Detection	Detect and locate the fault	Naïve Bayes classifier, SVM, K-means, SOM, PCA
Fault Classification	Determining source of the fault, Classifying the fault	Naïve Bayes classifier, DT, Transfer Learning
Outage Management	Detection of outage, Outage compensation	KNN, FFNN, SVM, DT, CF, K-means, SOM, Q-learning, PCA, MCA, DM, MDS

2.3.4 Edge Computing Applications

2.3.4.1 Edge Computing Hardware

The development of edge computing hardware enables machine learning on the edge level. Table 2.6 includes some of the representative edge computing devices. These devices can be classified

into three device types. The first type is the board devices. Board devices are standalone embedded computers that run machine learning algorithms independent of external devices. The second type is the accelerator devices. These devices cannot operate alone. Accelerator devices often act as an add-on to provide extra machine learning capabilities to embedded boards, personal computers, and other devices. The final type is smartphone chips. Smartphone chip manufacturers like Qualcomm, Hisilicon, Samsung, and MediaTek are pushing machine learning processing to mobile devices. Most of these chips rely on an AI accelerator to provide real-time machine learning processing capabilities.

2.3.4.2 Machine Learning on the edge

Machine learning applications on the edge layer can be separated into two major types. The first type aims to offload part or all of the existing functionality to the edge layer. This type of application is defined in this chapter as process offloading applications [77, 156–158]. The second type of application is referred to as sole functionality applications in this chapter. Sole functionality machine learning models often perform subtasks, which assist the main task on the cloud. The machine learning model of these subtasks is different from the model of the main tasks [159–162]. Table 2.7 summarizes all the works with different motivations for applying edge computing.

The motivation for process offloading applications is the limited resources of devices. The authors from [156] pointed out that low latency is essential for vehicle-to-everything applications. This work classifies vehicle-related applications into critical applications, high priority applications, and low priority applications. Critical applications include vehicle control systems, system monitoring, and accident prevention. These applications must be deployed on the very edge to the vehicle for a near-instant response. High and low priority applications include navigation and entertainment. These applications should be deployed on edge servers to enhance the computational capability of end user devices. This also ensures a relatively low latency.

The authors of [157] applied a similar offloading scheme to general machine learning web applications. The aim of this work is to offload computation power demanding machine learning tasks from embedded devices to an edge server. To achieve this, the edge device transmits a snapshot of the execution state before processing a machine learning task to the edge server. This method is independent of the type and model of the machine learning algorithm. However, the size of a snapshot is still enormous for embedded devices.

The authors from [77] further revealed that edge computing could also be used to protect user privacy. Their application uses a neural network to recognize certain objects from live streaming video. To protect user privacy, the first few layers of the neural network are offloaded to the edge servers. This also reduces energy consumption for the whole system, since processing is distributed among the network. However, as the users still need to send raw information to edge servers to be processed, privacy leakage remains an issue. This issue can be solved by directly deploying these first layers of the neural network to the end device. As a result, users only send

Table 2.6: Machine Learning Edge Computing Hardware.

Reference	**Hardware Series**	**Recent Model**	**AI Processor/Accelerator**	**AI Performance**	**Device Type**
[163]	Nvidia Jetson	Jetson AGX Xavier	512-core NVIDIA Volta GPU with 512 Tensor Cores	32 TOPs	Board
[164, 165]	Intel Neural Compute Stick	Intel Neural Compute Stick 2	Intel Movidius Myriad X Vision Processing Unit	4 TOPs	Accelerator
[166, 167]	Coral Dev Board	Coral Dev Board	Google Edge TPU ML accelerator coprocessor	4 TOPs	Board
[167, 168]	Coral USB Accelerator	Coral USB Accelerator	Google Edge TPU ML accelerator coprocessor	4 TOPs	Accelerator
[169]	Qualcomm Snapdragon	Qualcomm Snapdragon 855 Mobile Platform	Using CPU, GPU and DSP	Undisclosed	Smartphone Chip
[170]	HiSilicon Kirin	HiSilicon Kirin 980	Dual Neural Processing Unit	Undisclosed	Smartphone Chip
[171]	Samsung Exynos	Samsung Exynos 9820	Neural Processing Unit	Undisclosed	Smartphone Chip
[172]	MediaTek Helio P Series	MediaTek Helio P90	MediaTek APU 2.0	Undisclosed	Smartphone Chip

processed intermediate data to the network. All the works above only use edge computing primitively to offload computation requirements. However, machine learning by edge computing should leverage some unique properties of edge devices. The authors of [158] proposed a collaborative edge-centric learning method to train machine learning models. Each sensor contains a model that is trained locally using only data from that sensor. Training locally allows sensors to utilize contextual parameters to improve model accuracy. After training the local models, only the parameters of the models are sent to the sink from the sensors. This method reduces network overhead and energy consumption during training.

Different from the previous process offloading applications, sole functionality applications improve the performance of the system by performing a different subtask of the major task in the cloud. Earlier motivations are also related to the limited resources of devices. The authors of [161] utilized multiple filters, including CNN and SVM, to drop blurry and unwanted image data at the edge layer. The usage of filters reduces the processing power required on upper layers to create a training dataset for other applications.

Similarly, The authors from [159] also applied data cleansing on the edge layer to filter blurry images. Data cleansing is done by K-means in their food recognition system. Image segmentation is further applied as a data preprocessing method to reduce the load of the cloud server. However, the significance of this work is the utilization of locational data as a unique data type provided by edge devices. Furthermore, the authors used the locational data as a basis for collaborative recognition on the cloud layer.

To enhance localized service, the authors of [162] implemented network traffic prediction via LSTM on the edge cloudlets of a healthcare system. The purpose of this machine learning model is to predict bidirectional traffic between the cloud and the cloudlet to control data transmission rate and data caching frequency. These improve the quality of service and the reliability of data. As the LSTM model is deployed locally on cloudlets, the control decisions of the model are different between different cloudlets due to the different local network traffic.

Similarly, the authors from [160] also used machine learning to predict future sensor data. This is based on multi-variable regression and LSTM in their traffic monitoring system. These models are implemented on the edge servers to provide parameters for determining the quality of the video to be sent from the edge servers to the cloud. Therefore, this application aims to reduce network traffic by control data transmission from edge servers during non-critical events. The origin of these advantages is the increase of connectivity by introducing more edge servers to the system.

As machine learning applications on the edge attract much attention, the emergence of TinyML provides further advancement of these applications. TinyML combines embedded IoT technologies with machine learning [173]. It has the advantage of low bandwidth usage and latency like other edge computing applications [1]. On the other hand, TinyML applications aim to minimize energy consumption (below 1 mW). To deploy a machine learning model on such a low consumption device, model size also needs to be minimized. Balancing between model size and accuracy

Table 2.7: Motivation of Edge Computing.

Reference	Application	Edge Motivation	Application Type
[156]	Vehicle-to-Everything	Enhance computational capabilities Reduce latency	Process Offloading
[77]	Video Recognition	Process offloading Reduce latency Reduce energy consumption Protect privacy	Process Offloading
[157]	Machine Learning Web App	Process offloading	Process Offloading
[158]	Smart IoT Application	Reduce network overhead Reduce energy consumption	Process Offloading
[159]	Food Recognition	Data preprocessing Data cleansing Reduce latency Reduce energy consumption Location awareness	Sole Functionality
[160]	Traffic Control	Reduce network Traffic Increase scalability Ensure mobility Reduce latency	Sole Functionality
[161]	Graphical Expert System	Process offloading Data preprocessing Data cleansing	Sole Functionality
[162]	Healthcare System	Reduce latency Reduce network traffic Increase reliability Increase security	Sole Functionality

is a challenge for implementing TinyML applications [173].

2.3.5 Edge-Cloud Collaboration

In the traditional IoT architecture, machine learning algorithms on the cloud layer usually perform analytical tasks. However, novel applications are proposed utilizing the collaboration between

edge and cloud layers. Table 2.8 includes some edge-cloud collaboration methods.

A most common type of edge-cloud collaboration is the sole functionality applications from the subsection above. The healthcare system from [162] is an example. The system aims to classify and store data at different nodes of the cloud server. Data is collected from mobile devices and passed to the cloudlet layer. In the cloudlet layer, LSTM is implemented to predict network traffic. The prediction results are used for data transmission rate control and caching frequency control. Then, data is transmitted to an upper network layer. This layer utilizes a FFNN to classify traffic. Finally, these data are stored on the cloud according to the classified traffic types. In this application, the edge layers support upper cloud layers by completing subtasks. The result of the subtasks helps the cloud layer to perform the main task.

Edge assisted training is another type of edge-cloud collaboration. The authors from [161] used CNN and SVM to filter out images on the edge layer. This filter is to prevent corruption of the training on the cloud. Hence, it decreases the time required for an expert to create a training set.

The authors from [174] used federated learning to create an AE model for anomaly detection. A local version of the AE model is trained on every edge device using its local datasets. Then, the weights of these local models are transmitted to the cloud server and aggregated to form one AE model. This cloud level AE model is redistributed to the edge devices for local anomaly detection. As less data is sent from the edge to the cloud, this method reduces bandwidth demand during training and ensures that the training dataset is not corrupted due to data transmission. However, this method only considers one model across the system.

The authors of [175] extended training to multiple models. This is achieved with a machine learning model management module on the cloud server. This module accepts sensor data from the edge layer and uses these data to train different machine learning models. Then, the machine learning model selector selects and distributes a suitable model for every edge platform based on device performance and characteristics. This method optimizes network performance as the most suitable model is deployed for every device.

Another edge-cloud collaboration method is process offloading scheduling. The authors from [77] addressed that edge servers have limited bandwidth. Thus, scheduling of cloud process offloading should be implemented to avoid network congestion. The authors of [176] implemented a similar scheduling method on 5G networks. They use deep Q-learning to schedule server app migration on mobile edge servers. This method aims to provide users with an optimal quality of service. The authors from [177] incorporated cross-layer communication into process offloading decisions. In this work, end IoT devices can communicate both with Unmanned Aerial Vehicle (UAV) edge servers and satellite cloud servers. If the IoT devices loose connection with UAV edge servers, the IoT devices could offload their computation tasks to the satellite cloud. A deep actor-critic learning method is proposed considering energy consumption and network delay to solve this scheduling problem.

This section summarizes many machine learning algorithms, hardware and applications. The

Table 2.8: Applications Involving Edge-Cloud Collaboration.

Reference	Application	Collaboration Method
[77]	Video Recognition	Process offloading decisions.
[176]	5G Mobile Network	Process offloading decisions.
[177]	General IoT Application	Process offloading decisions.
[161]	Graphical Expert System	Data cleansing to aid training.
[174]	Anomaly Detection	Training using Federated Learning.
[175]	Indoor Condition Prediction	Dynamic model selection.
[162]	Healthcare System	Collaborating the results of different subtasks on different layers.

usage of machine learning from a network perspective are described. Machine learning applications in the physical layer and network layer are elaborated. Scheduling and management of different network resources and process are major applications of machine learning on these two layers. Then, for the cloud layer, the applications of machine learning that enable edge-cloud collaboration are illustrated. Edge computing aids cloud applications through process offloading and edge-only functions (sole functionality). However, this only shows collaboration in the application layer (Edge-Cloud). Collaboration between lower layers or cross-layer machine learning applications are still limited. Different methods of machine learning are widely adopted as frost prediction methods. Section 2.8 evaluates a few machine learning and deep learning-based frost prediction systems. Machine learning-based frost prediction methods are also adopted in integrated active protection systems (Section 2.10). The performance of RNN-based models as frost prediction methods is also assessed in Chapter 3. Moreover, the frost prediction methods in later Chapters are mostly based on ANN. Furthermore, ensemble learning is applied in Chapter 4 to aggregate the prediction results from multiple local weak predictors. These implementations demonstrate machine learning usage on the application layer. Finally, Chapter 6 presents frost prediction at the edge.

2.4 IoT Scalability

Universal scalability is discussed in this section. Universal scalability is separated into hardware scalability, network scalability and service scalability. Table 2.9 defines these different scalability concepts.

Hardware scalability is the ability of a piece of hardware to be extended to cope with different environmental, network, and service requirements. A common method for implementing hardware scalability is offloading part of the device functionality to a server [178, 179]. The authors of [178] proposed an architecture that extends device functionality through device virtualization. Additionally, this work demonstrates device virtualization in the case of a multi-protocol scenario. As a solution, virtual gateways are deployed on fog servers to process the packets received by

the end devices. However, adding functions of another functionality group (for example, adding image sensors to a transceiver device) still requires modification from the hardware level. To avoid modification from the hardware level, the concept of synthetic sensors is proposed [179]. Synthetic sensors can be separated into the device level and the server level. The device level is assembled by sensor tags capable of sensing data from multiple sensing dimensions. These sensing dimensions are low-level data types include vibration, audio, camera, temperature, humidity, air pressure, illumination, color, motion, magnetic field, and Received Signal Strength Indicator (RSSI). Then, low-level data is transmitted to the server level. On the server level, machine learning algorithms process these low-level data and convert them into valuable results to users. In conclusion, synthetic sensors create a platform with all the raw data types required and extend its functionalities through server-based machine learning analytics.

Network scalability is the ability to dynamically scale resources up and down to process the incoming IoT traffic. A common method to ensure network scalability in wireless sensor networks is clustering. The authors of [180] reviewed common clustering algorithms. Their work outlines clustering into processes of cluster head election and cluster formation. Cluster head election is the process of choosing cluster heads from wireless devices, and these cluster heads gather data from other members of its cluster and transmit it towards the base station [180]. After the cluster heads are elected, other wireless devices advertise themselves to the cluster heads and form clusters around these cluster heads to join the network [180]. Therefore, new devices can easily join the network with the cluster formation process. As a result, scalability is achieved with clustering.

The clustering techniques assume devices in the network are homogeneous. However, in an IoT scenario, devices are heterogeneous [180]. As a solution, intermediate fog devices are utilized [181]. Similar to the cluster heads, these fog devices gather information from the end IoT devices and transmit it towards a centralized server. Different to the wireless sensor network scenario, fog devices are not chosen by algorithms. These devices are specialized as an intermediate server. The authors of [181] pointed out that as a new IoT device joins the network, the device drivers and services can be distributed on the fog devices to achieve a simpler integration process. Therefore, fog servers increase the scalability of IoT networks.

The extensibility of network coverage affects the availability of network services to mobile users. The authors of [182] explored antenna-based coverage and capacity optimization in cellular networks. Their work is based on two major phenomena. The first phenomenon is that the tilting of mobile network antennas affects network coverage and capacity. The second phenomenon is that there is a tradeoff between coverage and capacity. These phenomena are caused by an increase in the power of the received useful signal in a cell and the reduction of signal coverage due to antenna tilting. On the other hand, the authors of [183] addressed energy efficient parent selection of mobile IoT nodes.

To ensure further coverage, scalability induced by antenna tilting, online and dynamic antenna configuration using reinforcement learning can be applied to cellular networks [184]. This method also belongs to the SON self-optimization functionalities [11]. Finally, to further extend network coverage, satellites are incorporated to provide network backhaul for IoT networks. The usage of

Table 2.9: Types of Scalabilty.

Reference	Type of Scalability	Definition
[178, 179]	Hardware Scalability	The ability of a piece of hardware to be extended to cope with different environmental, network and service requirements.
[180–182]	Network Scalability	The ability to dynamically scale resources up and down to process the incoming IoT traffic.
[186]	Service Scalability	The ability to incorporate new services into the existing IoT system.

satellite backhauls provides advantages of cost efficient, ease of deployment, avoidance of damage from natural disasters, seamless coverage, and reliability [185]. This could be part of the universal coverage solution.

Service scalability emphasizes the ability to incorporate new services into the existing IoT system. The authors of [186] defined scalability requirements of IoT applications as explicit control flow, decentralized interactions, the separation between control and computation, and service location transparency. This work also categorized IoT service interaction types into direct interactions, indirect interactions, event-driven interactions, and exogenous interactions. After the evaluation of the service interaction types with the scalability requirements, exogenous interactions are the only service interaction type, which satisfies all scalability requirements.

Exogenous interactions incorporate a coordinator to manage all service interactions with different devices and services. Therefore control is always managed by coordinators and is separated from service computation. From [186], this type of interaction is controlled with explicit control flow as the control flow is defined by the coordinators. Also, as a definition of exogenous interaction, the control is always separated from service computation. Furthermore, exogenous can be decentralized in a hierarchical manner. Finally, location transparency is provided by exogenous interaction because coordinators are controlling the service interactions, and location data is encapsulated during the process.

2.4.1 SDN Induced Scalability

SDNs bring programmability into traditional networks. Forwarding devices such as switches and routers can be virtualized in SDNs. This is achieved through the separation of control plane and data plane. As a result, SDNs simplify network management, minimize the limitation from hardware, and are easier to extend network functionality [5]. The advantages of SDNs could also be beneficial to manage D2D communication in 5G networks [187].

From Figure 2.14, an SDN architecture consists of the application layer, the control layer, and the data-plane layer. The application layer consists of software applications communicating with the control layer, the control layer process requests from the application layer and manage network devices, and the data-plane layer is network infrastructure such as switches and routers [5].

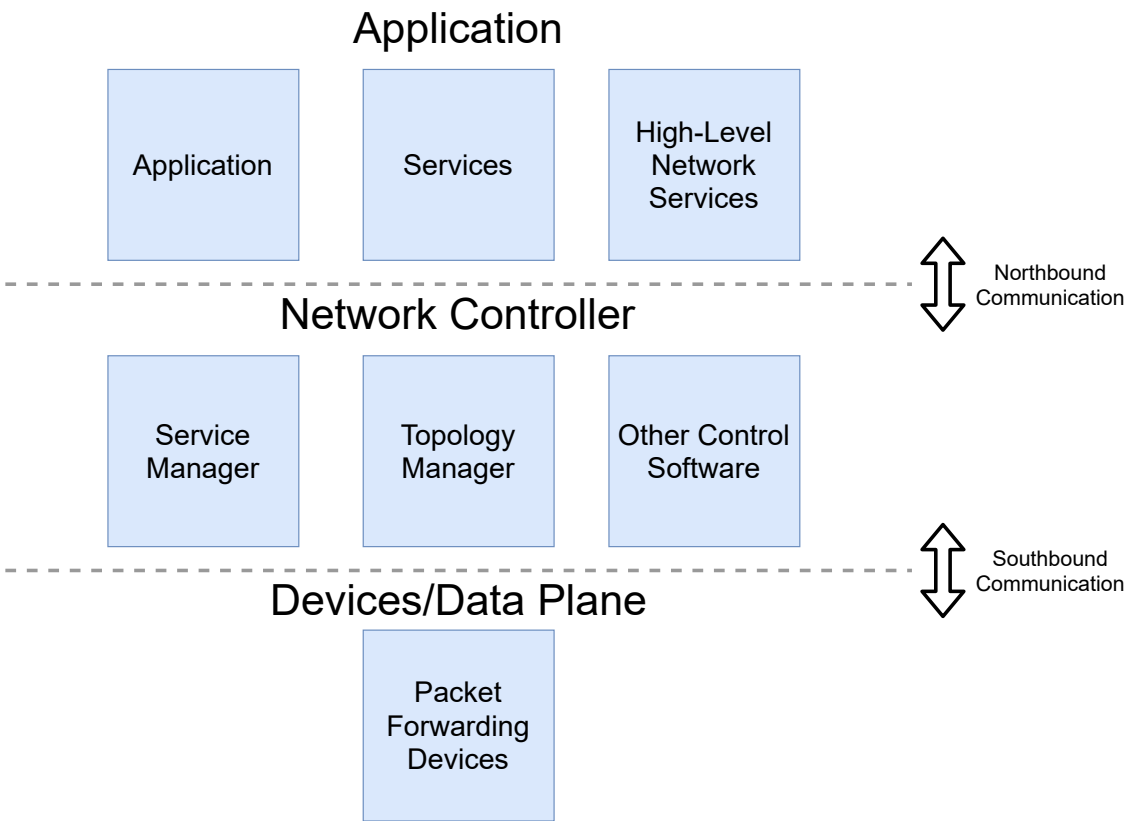


Figure 2.14: SDN Architecture. [5]

NFV is another technique that leverages service virtualization to increase network scalability. European Telecommunications Standards Institute (ETSI) defines a standard for NFV architecture (Figure 2.15) [6]. This architecture is assembled by the virtualized network functions (VNFs), the NFV infrastructure (NFVI), and NFV management and orchestration. NFVI includes the physical resource, which hosts VNFs as virtualized software implementations of network functionalities. Both NFVI and VNF are all managed by the NFV management and orchestration module. The advantages of the NFV architecture are reduction of hardware implementation costs, increasing flexibility and scalability by hosting VNFs on hardware, faster service modification due to software-based deployment, improved operational efficiency due to possible automation and operating procedures, improved power efficiency by planning and offloading workloads. NFV architecture is also able to create software interfaces to associate elements from different vendors.

The authors of [7] pointed out that SDN and NFV can benefit each other. SDN controllers can be treated as a VNF on the cloud providing flexibility to controller distribution. On the other hand, SDN provides its programmability to NFV, allowing communication between different VNFs. The combination of SDN and NFV further increases scalability. The authors of [7] also provided a software-defined NFC architecture that consists of the forwarding devices, the controller module, and the NFV Platform. From Figure 2.16, the forwarding devices are switches and routers from the data-plane layer of SDNs. These forwarding devices store forwarding tables to process a particular

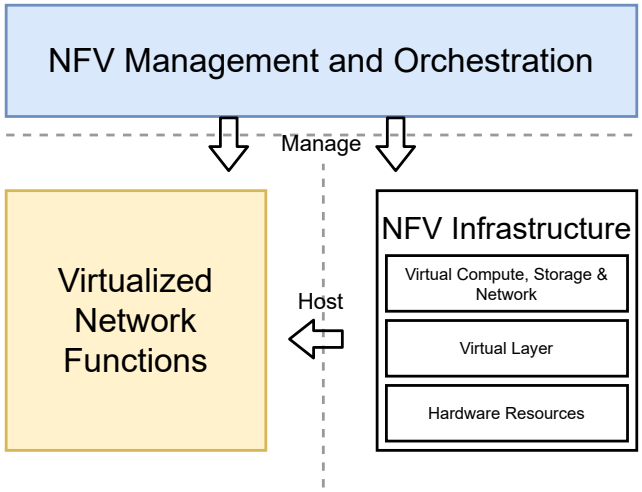


Figure 2.15: NFV Architecture. [6]

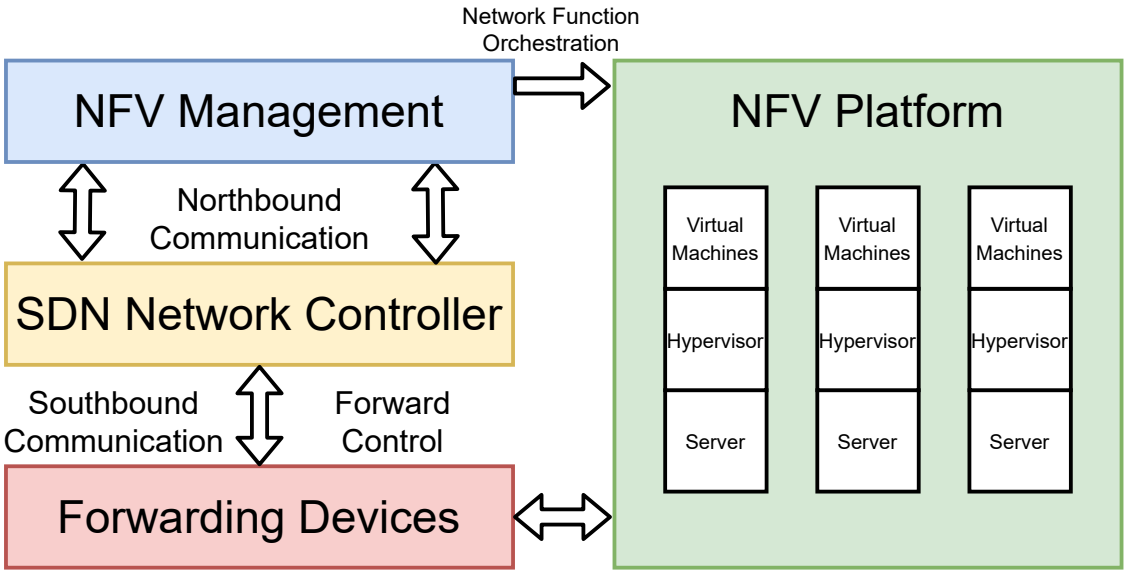


Figure 2.16: Software defined NFV Architecture. [7]

data packet. The forwarding tables are defined by the SDN controller. The SDN controller also controls NFV orchestration on the control module. Another function of NFV orchestration is to assign functions to the NFV platform, where servers host hypervisors supporting virtual machines running with the network functions [7].

The authors from [188] identified that in the environment of SDN and NFV, connecting and modification of virtual functions are complex due to multiple heterogeneous end-user demands and network parameters. Service function chaining could be a solution to reduce this complexity and optimize the use of resources. The authors from [188] also categorized existing service function chaining models into six optimization types as follows: network latency minimization, resource utilization optimization, cost minimization, power/energy minimization, service level agreement based optimization and quality of service based optimization. Finally, the authors of [7] provided a

vision of implementing service function chaining on the software-defined NFV Architecture. The optimal path of service chains is coordinated with the SDN controller fulfilling user requirements and resource constraints. Then, service chains are created from multiple VNFs, and data packets flow through the path of the service chains.

In this section, network and service scalability achieved with SDN and NVF are reviewed. The authors of [189] indicated the emerging network scalability issues trigger by the network management overhead in current networks with increasing size and dynamism. Autonomic or self-management of the networks (SONs) [189] could be a solution for these issues. On the other hand, IoT interoperability could be another solution to resolve scalability issues [12].

In Chapter 6, the implementation of pre-trained spatially generalized frost prediction models on edge nodes is a design that considers spatial scalability. Pre-trained generalized models can avoid the data collection time required for models based on local data when the frost prediction service needs to be expanded to a new location.

2.5 IoT Interoperability: Interoperability Between Standards

IoT networks are created with massive heterogeneous devices. The communication of these different devices is a key problem. To solve this problem, different standards are created to standardize the information exchanging process within IoT networks. The authors of [89] summarized all these standards and categorize them into communication, RFID, Data content and encoding, electronic product code, sensor, network management, middle, and quality of service. Apart from these protocols, there are also standards designed to fit the IoT use cases, such as, IoT6 [190]. With all these standards and protocols aiming for different scenarios, inter-communication between standards is an issue. This introduces interoperability problems of IoT. In Table 2.10, the authors of [12] classified interoperability problem into device interoperability, network interoperability, syntactical interoperability, semantic interoperability, and platform interoperability. The authors from [12] also aggregated different works and form seven approaches tackling the problem of interoperability. As the first approach, adapters and gateways are utilized as an intermediate bridge between different standards and specifications [12]. The intermediate device is compatible with multiple standards and specifications. Therefore, such a device can communicate with different IoT devices by converting messages between different protocols. However, this method assumes TCP/IP support on devices and does not account for the limitation of resources of IoT devices. Also, scalability is a problem as the message conversion process needs to be defined between all IoT protocols. The second approach is using a virtualized network overlay layer above physical networks. This approach supports end-to-end communication using different protocols. Unfortunately, scalability issues induced by different protocols persist.

The third approach in [12] consists of four different network technologies. The first technology is TCP/IP. Interoperability is implemented by embedding the TCP/IP stack on smart devices. Therefore, these devices can communicate with standard network protocols. The second technol-

ogy is SDN. This programmable network technology provides intelligence, efficiency, security, and scalability to IoT networks. This can also be achieved with NFV, where virtual networks separate network functions with the physical equipment. Furthermore, physical equipment can be shared between different network functions. The final technology is fog computing. Fog computing relies on fog servers to preprocess raw data from the end devices and preparing these data to be interoperable for other applications [12].

The fourth approach is using open APIs [12]. A commonly used example is the REST API. Open APIs provide standard methods to access data or services. This provides cross-platform and cross-domain interoperability. A future direction is a generic API for uniform resource access.

A service-oriented architecture is implemented above the network layer as the fifth approach to achieve interoperability. The aim of this architecture is to package the IoT device resource as standard services. Therefore, device data can be standardized into services, providing syntactic interoperability [12]. The IoT6 standard is an example of this approach. IoT6 is an IPv6-based service-oriented architecture that provides interoperability between heterogeneous system components [190].

The last two approaches to achieving IoT interoperability are semantic web technologies and open standards. Both of these approaches require a recognized organization to provide common definitions [12]. Semantic web technologies define a common understanding of the various entities. Once a common vocabulary of standard, data and format is agreed, semantic interoperability can be achieved. The final approach is the establishment of open standards. These standards are provided by recognized organizations to achieve interoperability with IoT networks implementing these standards. An example is the AllSeen Alliance, defining the AllJoyn for device interoperability and the oneM2M for platform interoperability [12]. The ISO also developed a framework (ISO/IEC NP 21823) for IoT interoperability [191]. They established standards on semantic interoperability and network connectivity.

The frost prediction systems proposed in Chapters 4 and 5 require different data types from different origins. There are local sensor data, external weather station data, and satellite image data. Platform interoperability is needed for the proposed systems to facilitate these data with different data structures and access mechanisms.

2.6 User friendly IoT

This section provides insights into the usability of IoT 2.0. The purpose is to create a vision of future IoT that aims to lower the entry barrier of IoT services for non-expert users. This vision starts with exploring the previous technologies of lowering entry barriers for non-expert users. These technologies are cloud computing-based services, which are infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). The usability of these technologies is created by increasing accessibility, increasing scalability and flexibility, the use of virtualization, reducing cost on maintenance, and standardization [192, 193]. Accessibility is created by the fea-

Table 2.10: Types of IoT Interoperability. [12]

Type of Interoperability	**Definition**
Device Interoperability	The exchange of information between heterogeneous devices and heterogenous communication protocols;The ability to integrate new devices into different IoT platforms.
Network Interoperability	Interact between different system accounting routing, resource optimization, security, quality of service and mobility.
Syntactical Interoperability	Interoperation of data structure in exchanged information.
Semantic Interoperability	Descriptions or understandings of resource, operational procedures, data models and information models between different entities.
Platform Interoperability	Interoperability required for barriers created by different IoT stacks consist of different operating systems, programming languages, data structures, architectures and access mechanisms for things and data.

ture of the cloud, where users can easily access the services through the Internet. Scalability and flexibility are induced by the virtualization of hardware and software resources [193]. Therefore, users do not need to directly configure these resources, as a standardized interface can reduce the complexity of resource configuration [192]. Finally, as maintenance is mostly done by the service providers, the cost of maintenance is reduced on the user side [192].

The authors of [194] offered the five features to ensure the usability of IoT systems. These features are Plug & Play, interoperability, the ability for remote control and monitor, cost effectiveness, and open source, open architecture. Also, by deduction from the cloud computing services, a standardized interface can increase the usability of systems and fulfill the features of remote control and interoperability. However, due to the heterogeneity of IoT devices, a solution is to adopt modularization. The authors from [195] proposed Internet of Things as a service (iTaaS). iTaaS utilizes service oriented architecture, which is built with modular and reusable service modules. Thus, this architecture reduces the time of service development, service deployment, and service configuration.

iTaaS only reduces complexity and interoperability issues for software deployment. Different from cloud computing services, IoT devices are heterogeneous and deployed in complex environments. Therefore, the customization of IoT devices is important to support different use cases [196]. The authors from [196] also emphasized that the modularization of IoT devices can reduce cost and complexity for non-technical personnel. Therefore, modularization increases the cost effectiveness and usability of IoT systems.

To reduce the cost of maintenance, IoT devices must operate in a self-organized, secure manner and avoid extra human intervention. The authors of [197] mentioned that self-organized IoT networks should contain the following features: cooperative communication model to support communication across different layers with suitable resource control, situational awareness to

monitor neighbor devices and faults, and automated load-balancing to extend the overall lifetime of the whole system. A possible solution is the SONs mentioned in the sections above. SONs provide machine learning-driven self-configuration, self-optimization, and self-healing functionalities [148].

Finally, as a vision towards the future, industry level standardization and efficient device deployment method is required for IoT systems. The authors from [198] pointed out that standardization drives standard testing and manufacturing procedures. Thus, this provides end users with more trust and confidence in IoT products. Another problem with current IoT systems is device deployment. Most work on IoT deployment focuses on software deployment and topology [194]. However, most devices are still deployed by humans. Novel device deployment methods should be invented to automate this process.

Chapter 5 modularized the proposed frost prediction framework. Users can replace the methods used in the modules to achieve an easier implementation of frost prediction and automated equipment control. The pre-trained spatially generalized models tested in Chapter 6 is a design to ensure the implemented edge nodes could achieve the feature of Plug & Play at new sites of operation without collecting local data.

2.7 Concepts of Frost

The second half of this chapter discusses the frost prediction methods in terms of the capability of real-time prediction as well as the requirements of relevant spatial and temporal resolution. In practical terms, and in addition to the obvious requirement of reliability and accuracy, desirable attributes of frost prediction methods must also include a reduction in the cost of frost protection and being able to provide crop managers appropriate warning time in order to implement effective response strategies [13,31]. The process of prediction and protection could be automated by leveraging CPSs with controllers controlling actuators based on sensor outputs, assuming the outputs have been preconditioned via some form of analytical processes in order to trigger a correct and timely response [84]. Therefore, and given CPSs by their nature conduct real-time operations [84], this research focuses on the short-term active frost protection methods, rather than the long-term passive frost protection methods. The primary aim of this part of the chapter is to identify current challenges and research gaps in automating real-time frost protection systems with minimal operational cost. The major contributions of this part are listed as follows:

1. Analysis of existing frost prediction algorithms and methods.
2. Summary of the existing work on active frost protection applications and methods.
3. Overview of protection systems leveraging frost prediction models.
4. Identification of current research gaps and future directions.

The definition of frost typically follows either the physical process itself, or the effect of its occurrence can vary in different literature. Through the former, frost refers to the phenomenon of ice crystal generation from dew or vapor [199], as "...the occurrence of an air temperature of 0 ^oC or lower, measured at the height between 1.25 and 2.0 m above soil level, inside an appropriate weather shelter." The second definition focuses on the damage of crops. The term "frost protection" is also ambiguous. Freeze damage or injury is caused by plants in an environment under a critical temperature, where water freezes within and damages the cells of crops [200]. As the words "frost" and "freeze" are often used as substitutes of each other and the term "frost protection" is used more frequently, "frost protection" generally means "freeze protection" [199]. Therefore, in this chapter, frost protection is defined as protecting crops from freeze damage or injury.

The authors of [13] categorized frost into radiation frost and advection frost (freeze) (Table 2.11). Radiation frosts occur in nights with calm wind and clear skies. During the event of a radiation frost, the temperature drops due to the natural radiation of heat [200]. Hence, temperature inversions occur, whereas advection frosts are formed without temperature inversions and in the presence of through wind. In addition, advection frosts are formed by a body of cold air moving to the site through wind [200].

Table 2.11: Environmental Characteristics of Radiation and Advection Frosts. [13]

Frost Type / Property	**Radiation Frost**	**Advection Frost**
Wind Speed	<5 mph	>5 mph
Cloud Coverage	Clear sky	Could be with cloud
Cold Air Thickness	30–200 ft	500–5000 ft
Temperature Inversion	Inversion develops	No inversion

2.8 Frost Prediction Methods

Frost prediction methods aim to detect an incoming frost event. In this chapter, frost events are categorized as classification methods and regression methods. Classification methods classify an input case into frost or no frost events for a future period of time, whereas regression methods predict the minimum temperature of a future period of time. As weather conditions could vary due to small spatial variations [31, 201], spatial resolution is essential for frost prediction. Also, real-time prediction and high temporal resolution prediction are both important to save the protection cost. The operational cost of active protection methods can be reduced as prediction resolution increases. This also gives the farmers more time to decide their tactics [31]. The rest of this section presents and discusses some relevant work on frost prediction with the three identified factors: instantaneity, high spatial resolution, and high temporal resolution.

Table 2.12: Frost Classification Prediction Methods.

Year	**Prediction Method**	**Input Data**	**Output**	**Performance**	**Real-time**	**High Spatial Resolution**	**High Temporal Resolution**
1976 [202]	Weather forecast	Temperature, Empirical experience	Indication of frost or no frost	N/A	**X**	**X**	**X**
1984 [31]	Weather forecast, Manual monitoring	Temperature, Dew point	Indication of frost or no frost	N/A	✓	✓	✓
1996 [203]	ANN	Daily max temperature, Daily min temperature, Humidity at 1900, Cloud cover, Wind speed at 1900, Wind direction at 1900	Min temperature $\leq$ 1.0 OR Min temperature $>$ 1.0	Rate of correct predictions: 88% - 94%	**X**	**X**	**X**
2009 [204]	SOM, ANN	Temperature, relative humidity, wind direction, wind speed and dew point	Indication of frost or no frost	N/A	**X**	**X**	**X**

Table 2.13: Frost Classification Prediction Methods (Cont.).

Year	**Prediction Method**	**Input Data**	**Output**	**Performance**	**Real-time**	**High Spatial Resolution**	**High Temporal Resolution**
2016 [205]	Logistic regression, DT	Min temperature, grass min temperature, dew point, Difference of max & min temperature, mean relative humidity, min relative humidity	Indication of frost or no frost	Probability of detection (Logistic Regression): 0.747 - 0.816. Probability of detection (DT): 0.731 - 0.866	**X**	✓	**X**
2017 [206]	Random forest	Temperature, relative humidity, solar radiation, dew point, wind speed and direction	Indication of frost or no frost	Overall success rate: 79% - 98%	**X**	✓	**X**
2018 [207]	Logistic regression	Temperature, relative humidity, weather station data	Indication of frost or no frost	True positive rate: 0.82 - 0.88	✓	✓	**X**
2019 [208]	ARIMA	Dry bulb temperature at 60 cm above ground at sunrise and sunset, Dew point at sunset, Wet bulb temperature at 60 cm above ground at sunset, Relative Humidity at sunset	Indication of frost or no frost	True positive rate: 0.60 - 1.00	**X**	✓	**X**

2.8.1 Classification Methods

Current classification methods focus on prediction of a possible frost event over a following period of time [31, 202–207]. Tables 2.12 and 2.13 summarize works on classification methods. A primitive method is described by [202]. The major contribution of the work is to measure the value created by frost forecasting. However, it also records farmers using temperature from the weather forecast with their empirical experiences to decide whether to switch on their protection heaters or not. This method is neither real-time nor high in temporal resolution as the prediction covers the whole night. As many farms scattering across a large county are using the same data source, this method fails to achieve high spatial resolution.

In [31], the farmers achieved real-time prediction with high spatial resolution and temporal resolution. However, their method uses a substantial amount of human labor. As the temperature reaches a preset point, farmers are informed by an alarm [31]. Then, the farmers would continuously monitor the temperature and dew point for the whole night to decide if a protection mechanism needs to be triggered [31]. This method matches all the preset criteria. However, it requires much manual intervention, and hence, it cannot be automated. On the other hand, the authors of [31] proves that protection mechanisms do not require to be switched on all night during a frost event. This gives motivation for developing a high temporal resolution prediction method to save operational costs on protection.

The authors of [203] leveraged a classification ANN to predict the occurrence of frost events for the coming night. The input of the model is temperature, humidity, cloud coverage, wind speed, and direction [203]. With these inputs, the model detects frost by labeling the input as "Temperature ≤ 1.0" and "Temperature > 1.0." The label "Temperature ≤ 1.0" indicates frost. The temporal resolution of this method is low because it is predicting frost occurrence for the whole night. Also, the spatial resolution of this method is questionable, since the data source is from Catania, Italy and it is unclear that these data are from multiple sources or not. Similarly, the ability for real-time prediction is also questionable since there is no experiment on the real-time inference of the model.

The authors of [204] aimed to create a frost prediction model with two stages of processing of the raw weather data. The first stage categorizes weather data into clusters using SOM. With the result of these clusters, a classification algorithm would classify new inputs as frost or non-frost events. However, this research only demonstrated results for the first stage of the study. Further evidence is required for the performance of the prediction algorithm.

Logistic regression and DTs are also leveraged for frost event prediction. These machine learning techniques are applied to compute the next day occurrence of a frost event using weather data collected from weather stations in Korea [205]. Prediction with a high temporal resolution still remains an issue as the prediction period is one-day long.

The authors of [206] improved the temporal resolution for classification methods to the next 12 hours. However, it is still not sufficient for creating a real-time protection system. Data are

collected from weather stations of the Maule region of Chile. Each model is trained using the dataset from only one region. Therefore, this method demonstrates a high spatial resolution.

The problem of temporal resolution still persists in [208]. However, it is still significant that the authors viewed the problem as a time series prediction problem using the Autoregressive Integrated Moving Average (ARIMA) model. This method allows the detection of all frost events in the test data. Unfortunately, the trade-off is high false alarms, which would increase the operational cost for the farmers to protect their crops.

2.8.2 Regression Methods

Earlier frost prediction regression methods are based on descriptive models of thermal radiation and heat convection [209–212]. In more recent work, neural networks demonstrated improvements in the results using the descriptive methods as a baseline [213]. Since the more recent methods have better results and the descriptive methods have already been analyzed in-depth, this chapter focuses on later models based on statistical methods and machine learning. However, descriptive methods can provide rapid and economically feasible predictions [214]. Therefore, the potential of these descriptive methods should be explored further in the future.

Tables 2.14 and 2.15 summarize works on regression methods. Most of the regression methods focus on predicting the minimum temperature of a future period of time with given historical or current inputs [207, 213, 215–219]. However, the method demonstrated in [220] is an exception. It is still considered as a regression method because the predictands are numerical values. This method uses Southern Oscillation Index (SOI) data, temperature and historical data of the date of last frost and the number of frosts to re-predict the date of last frost and the number of frosts of the coming year. Correlation analysis and linear discriminant analysis are conducted to find the relationship between SOI and the predictands. Furthermore, PCA and iterative clustering are used to study SOI phases. Unfortunately, this work cannot be applied in real-time as the results require years of historical data. Also, the temporal resolution of prediction is one year as this work predicts the date of last frost and the number of frost events of the coming year.

In [216], multiple models of temperature are produced considering the effect of the wind machine to predict temperature for different times of the day. This model predicts real-time with high spatial and temporal resolution as the data collected are from a 0.5 km^2 vineyard. However, this model fails to consider wind as a possible input factor as the study region is usually windless. Therefore, this method would be accurate for the study region of other windless regions. The model could fail in regions with winds as a significant factor affecting temperature.

The authors of [215] extended their work [203] from a binary classification model to a multi-class classification model with eight ranges of minimum temperatures. Since this is not merely predicting an occurrence of a frost event and even the model is a classification model, this method is considered as a frost regression prediction method. However, the limitations still persist. Spatial resolution and real-time prediction capability of this method remain questionable as the source of data are vague and there is no test conducted for real-time inference. Temporal resolution is still

Table 2.14: Frost Regression Prediction Methods.

Year	**Prediction Method**	**Input Data**	**Output**	**Performance**	**Real-time**	**High Spatial Resolution**	**High Temporal Resolution**
1996 [220]	Correlation analysis, Linear discriminant analysis, PCA, Clustering	SOI phases data, Temperature, Historical dates of last frost and number of frosts each year	Date of last frost, number of frosts	N/A	**X**	**X**	**X**
1997 [215]	ANN	Previous day min temperature, Previous day max temperature, Cloud cover, humidity at 1900, Wind speed and direction at 1900	A range of next 24-h min Temperature	Rate of correct predictions: 94%	**X**	**X**	**X**
2003 [216]	Linear regression	Elevation, Time of local sunset, Radiation received during the previous day, Distance to wind machines.	Temperature	Standard error: 0.24 - 0.60	✓	✓	✓
2006 [213]	ANN	Air Temperature, Soil Temperature, Relative Humidity, Wind velocity, Average Wind velocity, Max temperature of day, Min temperature of previous day, Daytime length	Min temperature of the night	Percent correct: 87.8% - 95.6%	✓	✓	**X**

Table 2.15: Frost Regression Prediction Methods (Cont.).

Year	**Prediction Method**	**Input Data**	**Output**	**Performance**	**Real-time**	**High Spatial Resolution**	**High Temporal Resolution**
2012 [217]	ANN	Max temperature, Min temperature, Average temperature, Max wind speed, Precipitation, Cloud cover, Moisture, Pressure, Humidity at 1900, Wind direction at 1900, Wind speed at 1900, 2 previous days min temperature, 5 previous days min temperature	Min temperature for the next 24 hours	True rate: 88%	**X**	✓	**X**
2018 [218]	ANN	Air temperature, Relative humidity, radiation, Precipitation, and Wind direction and speed	Next day min temperature	Accuracy: 0.97 - 0.99	**X**	✓	**X**
2018 [219]	Linear regression	Temperature, Dew point, Humidity	Min temperature of the night	N/A	**X**	✓	**X**
2018 [207]	Random forest	Temperature, Relative humidity, Weather station data	Next day min temperature	True positive rate: 0.75 - 0.9	✓	✓	**X**

low because the predictand is accounted for the next 24-hours. Also, prediction with two output classes demonstrates the highest accuracy. Therefore, extending towards multiple classes does not contribute to the improvement of frost prediction.

Similarly, ANN is also used to predict the minimum temperature of the night with daytime variables and previous day minimum temperature [213], but in a regression model. The model is built with high spatial resolution as data from individual weather stations is leveraged. However, the model is not tested for real-time inference. Also, the temporal resolution is limited, as it only predicts the minimum temperature of the night. On the other hand, an important result of this work is that the deep learning models have the potential to outperform the traditional descriptive models.

ANN is further leveraged to predict frost for the next 24 hours using rough sets [217]. The data is collected from a weather station located in Hunan, China. Therefore, the spatial resolution of the prediction is high. However, the method is predicting for the next 24 hours, which indicates low temporal resolution.

Similar work is done in [218]. ANN is applied on datasets from multiple weather stations of Central Chile to predict the next day minimum temperature with fewer parameters. Since the method is also predicting for the next day, low temporal resolution is still an issue.

The authors of [219] introduced the concept of WSN into frost prediction regression applications. They used WSN to increase the reliability of the data collection process. This ensures a high spatial resolution of prediction. Also, this system allows real-time inference, as the decision process is based on real-time data. However, the temporal resolution remains low as they are only predicting the minimum temperature of the coming night.

The authors of [207] also used WSN to predict frost occurrence. It performs both regression and classification methods using the random forest and logistic regression, respectively. Also, an external weather station is introduced to the temperature readings from WSN sensors. As WSN is used for data collection, the prediction method can conduct real-time prediction with high spatial resolution. Again, the temporal resolution remains low as prediction is for next day minimum temperature and next day frost occurrence. However, the significance of this work is using the Synthetic Minority Oversampling Technique (SMOTE), which significantly improves the performance of random forest and logistic regression models.

In conclusion, both classification and regression methods all predict with high spatial resolution as the data are collected from a sole weather station or using WSN. On the other hand, most systems are not tested for the capability of real-time prediction. Since most of the methods predict for the next day or the next 24-hour, the methods could potentially be extended for a higher temporal resolution. However, classification methods are hard to extend because, in the case of hourly or higher resolution prediction, the indication of frost events is complex compared to next day predictions. For example, as frost already occurs in the first hour of the event, in the second hour it is hard to indicate if the existing frost appears due to the environmental factor of the second hour or the first hour. On the contrary, an extension on regression methods is more straightforward

as the hourly minimum temperature can be measured with a temperature sensor.

2.9 Frost Protection Methods

Frost protection methods consist of passive protection methods and active protection methods [14, 15]. Passive protection methods are applied before the event of frost and are usually cheaper. However, passive methods might not provide sufficient heat to resist frost injury [15], whereas active protection methods are applied during the event of frost to preserve heat on the crops for protection. Active protection methods are usually expensive due to manual operation and maintenance costs [15]. Therefore, this chapter focuses on the exploration of automated systems to mitigate the operational cost of active protection methods. The rest of this section introduces both passive and active protection methods, but emphasizing on the works of active protection methods.

2.9.1 Passive Frost Protection Methods

Table 2.16 lists the advantages and disadvantages of some common passive frost protection methods. A common aspect of passive frost protection methods is that these methods need to be applied before the frost events [14, 15]. Hence, passive frost protection methods are not explored in this chapter, as this chapter focuses on real-time automated protection systems. On the other hand, passive methods are usually less expensive and effective against advection frosts [13]. Thus, passive methods could be a complement to the active protection methods.

2.9.2 Active Frost Protection Methods

Table 2.17 lists the advantages and disadvantages of some common active frost protection methods. Active frost protection methods are applied during an event of frost [15]. Therefore, active frost protection methods can be implemented on a real-time frost protection CPS. However, active protection methods are usually ineffective against advection frosts because of the high cost of large scale deployment and the lack of natural heat source in the environment [221]. The rest of this subsection demonstrates some works on active frost protection methods to summarize the current research trend.

The majority of works on active frost protection methods are related to sprinklers [222–225, 227] and driven by air disturbance technologies [17, 228]. The ultimate goal of most works on sprinklers is to improve the efficiency of water usage. To achieve this, heat convection mechanisms are studied to estimate efficient water application rates for frost protection [222,223]. Furthermore, choosing the suitable rate of application is important as over-irrigation would cause water logging issues and under-irrigation would cause potential frost injuries [227].

Some other works on sprinklers emphasize the comparison between different equipment setups. For example, in [224], settings of different sprinkler application rates with or without plant

Table 2.16: Common Passive Frost Protection Methods. [13–16]

Protection Method	**Protection Mechanism**	**Advantages**	**Disadvantages**
Site Selection	Avoidance of frost prone area. Choosing a site with soil in favor of heat transfer and storage. Choose higher spots to avoid dense cold air.	Could completely avoid frost.	Heat in soil might not be sufficient after a cloudy day.
Cold Air Drainage	Placement or removal of vegetation and other obstacles to control cold air drainage.	Can provide high degree of protection when the drainage pattern is known.	Could lead to erosion with the removal of plants and vegetation.
Plant Selection	Select frost resistant plants.	Significantly reduces the risk of frost damage.	Limited to certain crop types.
Canopy Trees	Enhance downward radiation	Efficient protection method.	N/A
Plant Covers	Reduce heat loss to the air from the crops.	Good for small plants	Deficient ventilation could cause diseases. Might not be effective without an external heat source. High labor cost for implementation and removal.
Soil Cultivation Planning	Cultivation releases heat from soil.	N/A	N/A
Irrigation	Wet soil allows better heat transfer and storage.	Can be used for other farming applications.	High installation cost.
Cover crops Removal	Increases direct radiation to the soil.	N/A	Might cause erosion.
Soil Covers	Warming the soil.	N/A	Covering with vegetive mulches might reduce head transfer into the soil.
Chemical Treatment	Activate cold resistance or delay bloom.	Effective method.	Not suitable to all crop types.

Table 2.17: Common Active Frost Protection Methods. [13–17]

Protection Method	**Protection Mechanism**	**Advantages**	**Disadvantages**
Heaters	Converting fuel to heat to replace crop heat loss.	Lower installation cost.	Expensive fuel cost. Not efficient as heat loss to the sky.
Wind Machines	Mix warm air above with the cooler air at the surface	Efficient usage of fuel.	High installation cost. Induce noises. Fan could be damaged during heavy wind or supercooled fog.
Helicopters	Moving warm air above towards the surface.	High area coverage.	High operational cost.
Sprinklers	Heat releases when water changes its state.	Low operational cost and labor requirement.	High installation cost. Require large volume of water. Some could be affected by the wind drift effect.
Artificial Fog	Smoke or fog traps radiating heat.	Effective for radiation frost	Only effective for a relative short time. Cause pollution.

Table 2.18: Works on Active Frost Protection Methods.

Year	**Protection Method**	**Research Aim**
1981 [222]	Sprinkler	Prediction of the application rate required to keep a leaf at 0^{o}C.
1986 [223]	Sprinkler	Investigation on the water requirement for frost protection.
1992 [224]	Sprinkler & Enclosure	Comparison of effect of protection between various settings of the enclosure and application rates of the sprinklers.
2002 [225]	Sprinkler	Comparison between efficiencies of microsprinklers and microsprayers.
2009 [226]	Electrically heated cables	Testing the effectiveness of the electrically heated cables.
2016 [227]	Sprinkler	Investigation on the effect of application rate towards the surface temperature of tea leaves.
2018 [17]	Wind machines, Selective Inverted Sink (SIS), Helicopters	Review on the control, effectiveness and working environment of air disturbance technology on frost protection.
2019 [228]	Wind machines	Investigation on the effectiveness of portable wind machines comparing to the stationary wind machines.

enclosure are compared to detect the setting with a higher temperature under the same environment. As a result, the setting with the enclosure and a high rate water application of 1.22 cm h^{-1} is proven to maintain the highest temperature. Another work [225] compares the efficiency between microsprinklers and microsprayers. Microsprinklers generally have better performance than microsprayers due to a higher rate of water flow. However, the difference is not significant with a temperature above 3^{o}C.

Other than sprinkler irrigation methods, air disturbance technologies are also widely studied. The authors of [17] provided a review of protection methods depending on air disturbance technologies. These methods include wind machines, selective inverted sinks (SIS), and frost protection helicopters. The review of these methods is conducted in terms of working principle, control factors, and selection criteria. As a result, the advantages of air disturbance technologies are a long-lasting period of protection and high effectiveness against radiation frosts. However, the common limitations are high initial cost, the presence of disturbing noise, dependency on the limited access of power supply, reliance on strong thermal inversion, and suitable wind direction.

To reduce the initial installation cost of wind machines, portable wind machines are proposed. A study [228] compares portable wind machines with stationary wind machines. As mentioned, portable wind machines do not require expensive initial installation costs and permanent installation. Also, some other advantages are less fuel consumption, less noise, and compatibility with different wind conditions. However, the portable wind machines are less effective than the stationary wind machines due to their lower engine power, narrower coverage angle, and lower height.

Finally, a novel method of frost protection is proposed in [226]. To mitigate the environmental requirements and water resource requirements on conventional heaters, wind machines and sprinklers, electrical heated cables are implemented to protect vines against frost damages. This method is effective as it significantly reduced loss from 46% to 13%. However, the use of electrical cables around vineyards would require extra care for the farmers during daily operations. Thus, potentially increasing manual labor cost. Moreover, large scale deployment would also incur a high manual labor cost.

2.10 Integrated Frost Prediction and Active Protection Systems

This section explores some implementations of frost protection CPSs (Table 2.19). In 1984, a primitive implementation of an automated frost protection system placed temperature sensors near the crops [31]. When the temperature reaches a preset point, a bedroom alarm as the actuator will be triggered to alert the farmers to switch on protection equipment. In more recent works, the triggering of protection equipment is automated with different frost prediction methods [226, 229–231]. However, most of these works still rely solely on temperature sensors to trigger the operation of protection equipment. A more sophisticated prediction algorithm could be implemented to provide a larger reaction window during periods with rapid changes in the environment.

Table 2.19: Works on Automated Frost Protection Systems.

Year	Protection Method	Alarm System
1984 [31]	Manual trigger of wind machine, Sprinklers, Heaters	Temperature sensors, Bedroom alarm
2008 [229]	Artificial Smoke	Temperature sensors, Fuzzy controller
2009 [230]	Sprinklers	Thermistors
2009 [226]	Electrical heating cable	Air temperature sensors, Timer switch
2012 [231]	Artificial cloud burner	WSN temperature sensors, Fuzzy controller
2017 [16]	N/A	WSN Temperature sensors, Public weather forecast service
2019 [232]	N/A	WSN sensors, Weather station service, Multivariate index
2019 [233]	N/A	WSN sensors, Weather station data

More recent works introduced external weather services, which calibrate with the deployed WSN sensors to provide better prediction [16, 232, 233]. Unfortunately, most of these works are still in an earlier stage. Therefore, there are limited results on prediction or integration with a protection method. As demonstrated by Table 2.19, these systems require integration with some actuators to become complete CPSs.

Another common issue of the current automated frost protection systems is low fault tolerance. As current systems are all unsupervised and depend on a small number of sensors and actuators, unavailability due to hardware defects or accidents on any of them could disable the whole system. As a consequence, the crops would suffer from frost damage. A more sophisticated system with high fault tolerance should be designed to eliminate this consequence.

2.11 IoT Communication Protocols for Frost Protection Applications

In this section, different IoT communication protocols are compared and evaluated in two stages. The set of criteria in the first stage is formed by evaluating IIoT requirements with the background of frost protection applications to choose the relevant factors. Then, the IoT communication protocols are compared and filtered using the result factors. Finally, protocols are further considered in a perspective of cost.

From [18], the requirement factors for IIoT networks are listed in Table 2.20. These factors are evaluated considering the operational environment of frost protection systems to be included as a prioritized factor. The evaluated frost protection systems are all utilizing single or a small number

Table 2.20: IIoT Requirements. [18]

Requirement Factor	Description
Latency	Can be improved by smaller packets and simpler protocols.
Reliability	Related to the amount of transmission errors.
Throughput	The amount of transmitted data in a fixed amount of time. High throughput is important for applications such as high resolution images or videos transmission.
Interference-robust capability	The ability against interference generated by other electrical equipment and communication systems.
Fading-robust capability	The ability against signal degradation due to wave reflection and scattering.
Energy efficiency	Energy efficiency is important for environment with limited number of stable terminals.
Communication range	The one-hop transmission distance.

of sensors for a relatively large field with limited amount of data transmitted [16, 226, 229–233]. Therefore, the system does not require a high throughput. Also, under the critical temperature, the freezing process is a (relatively) slow procedure up to 30 minutes with 10% of crop allowed to be eliminated for the purpose of thinning [31]. Thus, ultra low latency transmission is not required. Some latency and errors can be tolerated. With real-time inference of frost prediction methods, the system should still provide timely predictions. Overall, latency, reliability, and throughput are not critical factors for frost protection systems.

As agricultural IoT applications are mostly deployed in rural communities [234], the environmental characteristics of rural areas are influential for frost protection systems. The authors of [235] demonstrated that there are significantly less path loss and more extended transmission range in rural areas than suburban areas. Therefore, as frost protection systems are deployed in rural regions, interference-robust capability and fading-robust capability are not considered as priorities.

Since rural farmlands have a limited number of power terminals, and the wiring costs are high [234], the IoT nodes for agricultural applications have limited energy sources. Therefore, these IoT nodes need to be energy efficient to maintain the availability of the network [236]. Hence, energy efficiency is chosen as a factor of the criteria. On the other hand, the average European farm size is 16.6 ha [237], and the average Australian farm size is 4,331 ha [238]. Wireless communication covering such vast land requires a longer one-hop distance or an increase in the number of forwarding relays [239]. Since this affects the coverage of the network and the number of forwarding relays as a cost factor, communication range of IoT protocols is also evaluated. In the next paragraph, the energy consumption and communication range of some common IoT protocols are compared.

From Table 2.21, medium to high energy consumption protocols include Bluetooth, Wi-Fi,

WiMAX, Cellular. Operation with these networks result with a lower life span of the application [236]. Hence, these protocols are not suitable. On the other hand, protocols with low energy consumption often operate with a limited communication range around 100 - 200 m [19–21]. Therefore, more relays are required to forward the signals to the sink [239]. The Low Power Wide Area Network (LPWAN) protocols (LoRaWAN, SigFox and NB-IoT) are the only low energy consumption and long communication range protocols that are evaluated. Therefore, LoRaWAN, SigFox and NB-IoT are further analyzed with cost factors in the next stage.

Table 2.21: Comparison of IoT Communication Protocol Energy Consumption and Communication Range. [19–22]

Criteria / Protocol	**Energy Consumption**	**Communication Range**
6LoWPAN	Low	Short (10 - 100 m)
Bluetooth	Medium	Short (10 - 100 m)
ZigBee	Low	Short (10 - 100 m)
RFID	Low	Short (up to 200 m)
NFC	Low	Short (<1 m)
Z-Wave	Low	Short (30 - 100 m)
Li-Fi	Low	Short (around 10 m)
Wi-Fi	High	Short (1 - 100 m)
WiMAX	Medium	Long (<50 km)
Cellular	High	Long (several km)
LoRaWAN	Low	Long (5 - 30 km)
SigFox	Low	Long (10 - 40 km)
NB-IoT	Low	Long (1 - 10 km)

The authors of [22] split the cost of LPWAN protocols into spectrum cost, deployment cost, and end-device cost. From Table 2.22, NB-IoT is the most cost-ineffective in all three types of cost. Also, as LTE cellular coverage is not available for some farms [22], NB-IoT is not suitable for frost protection applications. If NB-IoT is not considered, the deployment cost would be the most significant for other LPWAN protocols. Private local networks can be deployed with cheaper LoRaWAN gateways instead of expensive base stations [22]. This significantly reduces deployment costs. From a cost perspective, among the assessed LPWAN protocols, LoRaWAN should be the most suitable for frost protection applications.

Table 2.22: Costs of LPWAN Implementation. [22]

Cost / Protocol	**Spectrum Cost**	**Deployment Cost**	**End-device Cost**
SigFox	Free	>4000€ per base station.	<2€
LoRaWAN	Free	>100€ per gateway. >1000€ per base station.	3 - 5€
NB-IoT	>500 M€ per MHz	>15000€ per base station.	>20€

2.12 Current Limitations and Future Directions of Frost Systems

This section provides an overall discussion on frost prediction methods, frost protection methods, and integrated frost protection systems. After outlining the common aspects of the current state, limitations are concluded to lead to possible future implementations. Several limitations are avoiding the construction of effective, real-time, and automated frost protection CPS.

Model accuracy is limited by the quality of local historical data From Tables 2.12, 2.13, 2.14, and 2.15 most prediction methods leverage machine learning techniques to build the prediction models. Although machine learning methods demonstrate results with high accuracy, their dependency on historical sensor data could be a limitation towards higher accuracy. The demand for higher accuracy is valid because any false-negative results of frost prediction could induce a substantial loss to the agricultural sector.

This limitation on model accuracy is consistent with the constraints of machine learning models. As machine learning models are highly dependant on data, data quality is often influential to the performance of the model [94]. In most systems, data is collected by sensors. Therefore, data quality is highly bounded by the characteristics of the sensors. In an IoT context, sensor data could arrive with noises, errors, and discontinuities [71]. These data could corrupt the dataset and result in an inaccurate machine learning model. A possible solution would be applying various data cleaning techniques on the training dataset to remove the noises, eliminate the errors, and patch the discontinuities [240].

Another limitation of machine learning models is generality. Machine learning can provide models of the patterns from the training dataset with high generality. However, it cannot provide an accurate output beyond the patterns of the training dataset [241]. This phenomenon also applies to deep learning models [242]. The issue of generality could appear in both spatial and temporal dimensions. In the spatial dimension, as most machine learning models are built with local data, it could only provide accurate results within a local scope. Moreover, most model building and testing processes are also conducted with local data. Therefore, the generality of models is not confirmed at other locations. In the temporal dimension, since the patterns of future climate

change are unpredictable [243], models based on climate data from earlier years might decrease in accuracy. Possible mitigation techniques include rebuilding the models with a recent dataset and application of a self-adaptive learning model such as reinforcement learning.

Prediction models have low temporal resolutions According to Table 2.12 and 2.14, the current frost prediction methods often predict the occurrence of any frost for the next 12 and 24 hours. Therefore, the protection equipment needs to be switched on for 12 to 24 hours to minimize the risk of frost damage. However, by the results from manual observations [31], the duration of a frost event could be shorter. Thus, the operational time of frost protection equipment could be reduced to save the operational cost. A solution provided by [31] is manual monitoring, which would incur extra human labor costs. To eliminate this human labor cost, a potential improvement would be increasing the temporal resolution of prediction to hourly or even minute by minute prediction. As a result, this improvement could achieve the automatic operation of protection equipment with fewer operational hours.

Prediction models are often not tested in real-time scenarios Frost prediction in the agricultural sector allows farmers to plan their tactics to protect the crops and retain psychological comfort [31]. However, current frost prediction models demonstrate limited evidence of operation in a real-time scenario. To construct an operational cost-efficient protection system, real-time prediction results with the high temporal resolution are required. As a result, an efficient operation scheme can be generated by the system. In conclusion, tests for real-time prediction models leverage live WSN data acquisition should be conducted to confirm the capability of real-time predictions.

Active frost protection methods are vulnerable against advection frosts [221] Currently, large scale deployment of an active frost protection method is not economically feasible [221]. However, from Table 2.18, most research of active protection methods lay in the field of sprinklers and wind machines. These methods are vulnerable to advection frosts. An economically feasible active protection method could be a future research field to eliminate the vulnerability. This method could benefit from passive protection methods to reduce the cost and increase effectiveness against advection frosts [13]. Also, the potential of some chemical solutions has not been fully revealed. Ice nucleation inhibitors protect crops from frost damage by limiting the ice nucleation process. This is achieved through the inhibition of bacterial ice nucleating agents [244, 245]. The chemical solutions could be applied using the existing irrigation systems or sprayers to reduce the manual labor required.

Current frost protection systems have low fault tolerance Most frost protection systems from the works of Table 2.19 rely on only one or a small number of sensors and actuators. These systems would have a low tolerance for any system fault. Consequently, this would induce inaccurate decisions and even unavailability of the whole system. The faults can be classified into device faults

and network faults [246]. Device faults are generated by malfunctioning nodes and sinks, whereas, network faults are originated from the outage of network connections between devices [247]. The cause of these faults spans across the hardware layer, the software layer, the network layer, and the application layer [247]. The authors of [247] separated fault tolerance mechanisms into three categories, including redundancy-based, clustering-based, and deployment-based mechanisms [247]. Redundancy-based techniques are redundancy of data, path, reports, and nodes. As current frost protection systems operate with a few sensors and actuators, redundancy-based methods can increase the reliability of the system [248]. However, a trade-off between cost and reliability needs to be further evaluated [248]. The second type of fault tolerance technique is related to clustering. Clustering is a technique that increases the overall network lifespan by utilizing local cluster heads [249]. Clustering-based fault tolerance mechanisms could be applied to minimize the fault during cluster head selection [248]. This mechanism might not be active on current frost protection systems with a few nodes, but it can be applied to future systems implementing node redundancy. The final type of fault tolerance mechanism is the deployment-based mechanism [247]. Deployment-based mechanisms focus on topology control to adapt the network to the changes of node condition, noise, and interference [247]. These mechanisms can be applied to increase the reliability of the system [247], thereby reducing the possible economic loss of frost damage induced by system errors.

A final future direction would link back to the aim of the chapter. According to Table 2.19, some of the automated protection systems are lacking integration with a sophisticated prediction method, while other methods require actuators for frost protection. Therefore, the final future direction is the integration of a sophisticated prediction method leveraging sensor data with a protection actuator to form a real-time accurate frost protection CPS.

2.13 Summary

This chapter provides the technical background and first motivations of this thesis. Recent IoT architectures and the four dimensions of IoT 2.0 have been discussed as the basis of frost prediction and protection systems in this thesis. In response to **Research Question 3**, these discussions have revealed that IoT architectures can act as reference architectures to frost CPS systems. Machine learning intelligence lies on the application (intelligence) and edge layers of the IoT (CPS) architecture to predict frost and provide protection decisions. IoT Scalability of frost prediction systems is improved in later chapters with a spatially generalized prediction model. In addition, IoT platform interoperability supports the integration of data from multiple sources. As the final relevant dimension of IoT 2.0, the design idea of user friendly IoT leads to the modularization and standardization of frost prediction and protection systems. The second half of this chapter reviews recent frost prediction methods, frost protection methods, integrated protection systems, and their communication protocols. The limitations induced by historical data dependence, low prediction temporal resolution, non-real-time response, vulnerability against advection frost, and low fault tolerance are revealed as a reply to **Research Question 1**. **Research Question 2** is answered

in two stages. Firstly, the frost regression prediction methods are in favor over the classification methods due to the different threshold temperatures of different plants. Secondly, ANN-based models demonstrate the highest accuracy in the literature reviewed for regression methods. New research questions have emerged with the response to these previous questions. The limitations of low prediction temporal resolution and non-real-time response lead to **Research Question 4**: How to achieve near real-time frost prediction? This question is acknowledged in the next Chapter (Chapter 3). **Research Question 5** reflects on the limitation of low system fault tolerance. The spatial interpolation-based designs in Chapters 4 and 5 address the concerns of **Research Question 5**.

2.14 Review on Post-thesis Frost Prediction

This section summarizes some works on frost prediction after the design and implementation of the proposed systems in this thesis. Similar to the above sections, these works have been separated into classification methods (Table 2.23) and regression methods (Table 2.24).

In [250], a zoning tool is created to predict the risk of frost in forest plantations. Instead of relying on rapidly changing measurements (e.g., temperature, humidity), the tool is based on geological coordinates, elevation, relative altitude and hydrography. The zones provided are permanent despite the changing environmental conditions. Therefore, this method is unsuitable for real-time, high temporal resolution predictions. However, this tool is still significant for passive frost protection. During the planning phase of plantations, frost-sensitive crops can be planted in low-risk zones. The authors applied random forest, SVM and ANN. The results show that random forest has the highest accuracy and ANN has the lowest accuracy.

Similar to [250], the authors of [251] included a physiographic description of the study site with geological coordinates, slope, aspect, elevation, and curvature as prediction inputs. In addition to the physiographic factors, environmental readings (minimum temperature, relative humidity, wind velocity, and sunshine hours) are included as climatic factors for prediction. These variables are used with SVM and ANN models to predict frost occurrence in five classes (no frost, mild frost, moderate frost, severe frost, serious frost). Since this prediction is based on daily data, this method can not provide real-time predictions with high temporal resolution. The accuracy of the SVM predictor is higher than ANN. However, the accuracy of the ”no frost” classification of ANN (0.500) is higher than SVM (0.375). This shows fewer false positives of ANN than SVM. Overall, the accuracy of both methods needs further improvement.

The authors of [252] focused on prediction with environmental readings. Hourly prediction models of frost occurrence are created with logistic regression, DT, random forest, and SVM. Among these classification models, random forest and SVM demonstrate higher accuracies than the other models. These two models are extended with the initial frost occurrence time and minimum temperature at night. These new learning variables increased the model accuracy to over 93%. However, as the new variables can not be obtained with hourly observations, the model pro-

vides daily frost occurrence prediction for the next morning. In conclusion, the extended models cannot provide real-time predictions with high temporal resolutions.

In [253], the model inputs are all environmental and meteorological parameters. Frost occurrence probability of 1–5 days after the prediction is computed with logistic regression and ANN. The accuracy of both models is below 0.72, with a high false alarm ratio of over 0.38. Similar to the models before, the temporal resolution of prediction is low.

The authors of [254] introduced graph neural networks into frost prediction. A spatial-temporal graph is constructed to predict frost occurrence in six-hour intervals. Spatial-temporal relationships between the experiment site and 10 weather stations are modeled. The authors implemented two versions of the model. There is a classification model predicting frost occurrence probability and a regression model predicting the minimum temperature. Both models have low temporal resolutions.

In [255], nightly minimum temperature maps are generated using only one temperature sensor mote in a farm. The map is generated by computing the difference in terrain between the temperature sensor and the target location. The difference in terrain is measured by the difference in elevation, and the difference in standard deviation of elevation within a radius of 700 m. As this work provides nightly maps, the temporal resolution of prediction is relatively low. This work is still significant because it provides a cheap zoning method to indicate zones with high frost risks and contributes to passive frost protection methods.

Compared to the models reviewed in the above sections, the post-thesis frost prediction works also focus on low temporal resolution predictions. Low temporal prediction resolution has been viewed as a limitation in this thesis and improved in the next chapters of this thesis towards minute-wise predictions. Most of these works also rely on a single sensor node, which suffers from low fault tolerance. Overall, **Research Questions 4** and **5** can still be asked to determine the limitations in most post-thesis works. However, there are also improvements. The graph neural network [254] is the newly introduced model type in frost prediction. As multiple data sources are required for graph neural networks, the fault tolerance of the prediction system is enhanced. On the other hand, there are model-based solutions to aid passive frost protection [250, 255], which is a gap not covered by this thesis.

Table 2.23: Post-thesis Frost Classification Prediction Methods.

Year	**Prediction Method**	**Input Data**	**Output**	**Performance**	**Real-time**	**High Spatial Resolution**	**High Temporal Resolution**
2021 [250]	Random forest, SVM, ANN	Longitude, Latitude, Elevation, Relative altitude, Relief orientation, and Hydrography distance	Frost occurrence probability	Accuracy: Random forest: 0.91; SVM: 0.76; ANN: 0.67	**X**	✓	**X**
2021 [251]	SVM, ANN	Longitude, Latitude, Elevation, Slope, Aspect, Curvature, Minimum temperature, Relative humidity, Wind velocity, Sunshine hours	Frost damage as five classes	Average accuracy: SVM: 0.7929; ANN: 0.7129	**X**	✓	**X**
2021 [252]	Logistic regression, DT, Random forest, SVM	Temperature, Subzero duration, Precipitation, Wind speed, Humidity, Snowfall, Three-hourly snowfall, Ground temperature, Initial frost occurrence time, and Night minimum temperature	Frost or no frost	Accuracy: Random forest: 0.765, 0.9416; SVM: 0.771, 0.9361	✓	✓	✓
2021 [253]	Logistic regression, ANN	Max and Min temperature, Wind speed, Precipitation, Sunshine hours, Cumulative pan evaporation, Morning relative humidity, and Afternoon relative humidity	Frost occurrence probability 1-5 days after	Accuracy: Logisti regression: 0.75, 0.81; ANN: 0.70, 0.72	**X**	✓	**X**
2022 [254]	Graph neural network	Latitude, longitude, temperature, humidity	Frost or no frost in six-hour intervals	Precision: 0.891	**X**	✓	**X**

Table 2.24: Post-thesis Frost Regression Prediction Methods.

Year	**Prediction Method**	**Input Data**	**Output**	**Performance**	**Real-time**	**High Spatial Resolution**	**High Temporal Resolution**
2021 [255]	Multivariate adaptive regression splines	Elevation difference between logger and weather station, Standard deviations difference of elevation within a radius of 700 m between logger and weather station	Nightly minimum temperature maps	RMSE: 0.72-1.61^{o}C	**X**	✓	**X**
2022 [254]	Graph neural network	Latitude, longitude, temperature, humidity	Minimum temperature in six-hour intervals	RMSE: 5.42^{o}C	**X**	✓	**X**

3

Minute-wise Frost Prediction: An Approach of Recurrent Neural Networks

3.1 Introduction

This chapter focuses on predicting the condition of future frost damages to plants. A near real-time prediction system based on local sensors has been designed. The potential of RNNs in frost prediction is explored in this chapter. The standard RNN has issues such as gradient explosion and gradient vanishing [112]. To address these issues, LSTM and Gated Recurrent Unit (GRU) are proposed as variants of the RNN [256]. This chapter leverages RNN, LSTM, and GRU models for frost prediction. A minute-wise frost prediction system has been created to address **Research Question 4**. Most of this chapter has already been published as the journal paper "*Minute-wise Frost Prediction: an Approach of Recurrent Neural Networks*," in *Elsevier Array* [257].

In recent years, the Internet of Things (IoT) technologies have been widely applied in the field of agriculture to provide real-time monitoring and actuation services [258]. There are also a few IoT-based frost protection systems. However, most of these frost protection systems rely on thresholds of real-time sensor readings to trigger the frost protection equipment [34]. The effect of these simple mechanisms is limited compared to the accuracy of the prediction algorithms [34]. Therefore, this article considers a few factors related to the future deployment of frost prediction algorithms. These factors include model processing speed, long-term accuracy and data availability. Since system resources are limited for IoT systems, the model should require a faster processing speed [259]. Also, IoT systems should eliminate extra human interventions [258]. Therefore, the deterioration of model accuracy over time should be minimum to ensure manual updates to

IoT nodes are infrequent. Finally, as most frost prediction models depend on on-site historical data [34], data availability is important when creating these models. Since a large amount of historical data cannot be assumed to be available at all sites, our set scene assumes that only a small amount (three months) of data is available to minimize the data collection time for new models.

3.1.1 Related work

In Chapter 2 [34], frost prediction methods are categorized as "classification methods" and "regression methods." Classification methods predict the occurrence of frost as a percentage in a future time, whereas regression methods predict the minimum temperature in a future period [34]. Both methods rely on climate data as the model input. Since the frost resistances for different species of plants are different [260], frost regression prediction methods are proposed in this chapter to provide the farmers future environmental insights and provide a more generalized solution avoiding the differences between individual plant species.

There are a few existing frost regression prediction methods. In [216, 219] and [207], traditional machine learning methods are leveraged to predict temperature or minimum temperature in the next day or night. Random forest models are used in [207] to predict next-day minimum temperature with temperature and humidity inputs. Linear regression is used in both [216] and [219]. Environmental parameters such as temperature, dew point, and humidity are inserted as model inputs in [219]. On the other hand, to consider the effect of wind machines, the authors of [216] introduced the distance to wind machines along with elevation, time of local sunset, and radiation received during the previous day as input parameters.

Apart from the traditional machine learning models. ANN with fully connected layers, as a deep learning model, can also predict future minimum temperatures [213, 217, 218]. Models in [213, 217] and [218], predict minimum temperature in the next 12–24 hours as a numerical value. These three works all implement prediction models with air temperature, relative humidity, precipitation, wind direction and speed. However, [213] also includes daytime length, daytime maximum and minimum temperature to support night temperature predictions with a daytime baseline. In [217], precipitation, cloud cover, moisture, and pressure are included as model inputs. The authors also considered humidity and wind velocity at 19:00. The authors of [218] predicted next day minimum temperature with fewer input parameters, but introduced radiation to build their prediction ANN.

The above machine learning and deep learning models all predict frost conditions in the next 12 - 24 hours [207, 213, 216–219]. Therefore, in extreme conditions, protection equipment might need to be switched on for 12 - 24 hours to ensure zero frost damage when solely considering model predictions. However, by constant manual observations, the operational time of protection equipment could be reduced [34]. Hence, to reduce the operational time automatically, the major aim of this chapter is to implement minute-wise next hour minimum temperature prediction for frost prediction. Also, as mentioned in the above paragraphs, the potential of RNN-based models (RNN, LSTM, GRU) are explored to solve this prediction problem. The performance of different

RNN-based models is also compared with each other in this chapter. In conclusion, the major contributions of this chapter are presented as follows.

1. Application of an RNN-based frost prediction method.
2. Increasing the prediction frequency from once per 12–24 hours for the next day or night events to minute-wise predictions for the next hour events.
3. Comparing the errors of different RNN-based models for frost prediction.
4. Test the processing time required for training and inference for RNN models with different settings.

The rest of this chapter is arranged as follows. Section 3.2 describes the methodology and experiment settings with the study area, data processing procedures, and experiments. Experiments include comparing temperature prediction models (ANN, RNN, LSTM, GRU), analyzing different RNN model settings, and the performance of predicting minimum temperature with other frost-related parameters. Then, the experimental results are discussed in Section 3.3 and lead towards limitations with open challenges. In the end, Section 3.4 summarizes the whole chapter.

3.2 Methodology

This section explains the methodology and settings of the experiments. The study area is firstly explained, followed by the data preprocessing procedures. Then, model construction and testing leveraging the preprocessed data are described. Finally, the experiment processes are summarized.

3.2.1 Study Area

The study area is located in New South Wales (NSW) and Australian Capital Territory (ACT) of Australia. In the study area, datasets from 30 different weather stations are obtained. Figure 3.1 is a map of the study area with the weather station locations and IDs. Also, a list of location coordinates of the weather stations are presented by Tables in Appendix A [261]. All these raw datasets used in this chapter can be acquired from the public weather station directory service hosted by the Bureau of Meteorology (BOM) of Australia [261].

This study is focused on the months June, July, and August of winter [262, 263]. Minute-wise climate data of these winter months in years 2016 and 2017 are extracted from the 30 weather stations in the study area. After extracting these data, they are further processed to be prepared for model construction and testing. These procedures are discussed in the next subsection.

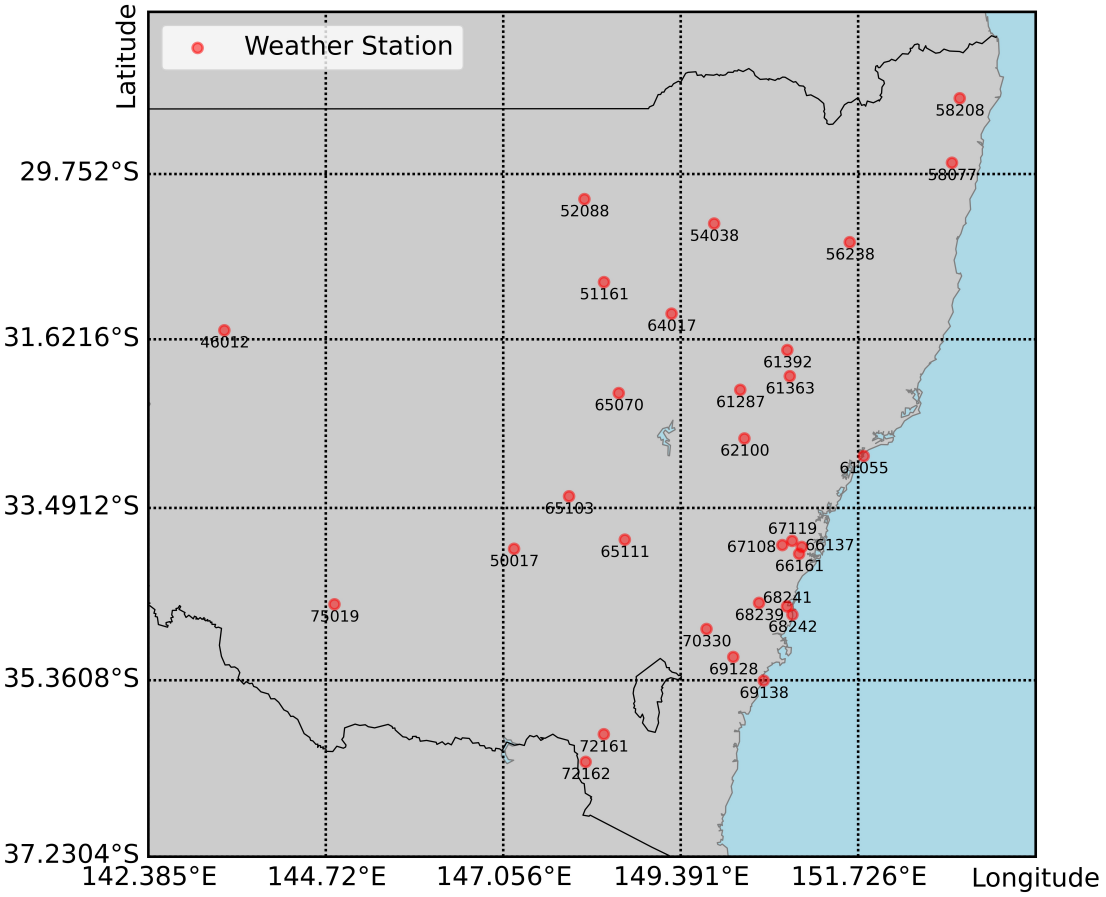


Figure 3.1: Weather Stations with ID

3.2.2 Data Preprocessing

The raw data needs to be preprocessed before model construction. After extracting the required columns (Timestamp, Air Temperature, Dew Point, Relative Humidity, Wind Speed, Wind Direction) from the raw datasets, seven more steps of data preprocessing are performed. These seven steps belong to two phases. In phase 1, the raw data is processed to transform useful features and handle empty values. The outputs of phase 1 are saved for reuse in the next phase. In phase 2, the outputs from phase 1 are inputted and modified to fit a specific target model and model setting. As an example, Figure 3.2 outlines the changes of data structure during data preprocessing, using an RNN with a sequence length of 20 as an example. The red figures indicate a change to that data in that step.

Phase 1 consists of five steps and starts with transforming wind speed and direction into features of "N-wind" and "E-wind". These two new features represent wind speeds toward the north (N-wind) and east (E-wind) directions. This transformation is done to prevent the possible errors generated by the wind direction values (0 - 359) [264]. The original wind direction data from BOM is the bearing of the direction the wind is originated [265]. Therefore, to begin the conversion, the original wind direction (met) is changed to the wind blowing direction (deg) by reversing the direction (Equation 3.1).

$$deg = \begin{cases} met + 180°, & \text{if } met < 180° \\ met - 180°, & \text{if } met \geq 180° \end{cases} \tag{3.1}$$

Then, with the wind speed (v) and the wind blowing direction (deg), magnitudes of wind vector toward north (v_N) and east (v_E) are obtained through Equation 3.2 [264].

$$v_E, v_N = v \times sin(deg), v \times cos(deg) \tag{3.2}$$

In the second step of data preprocessing, the minimum temperature is constructed as the prediction target. The minimum temperature is simply constructed by getting the minimum temperature value in the next 60 minutes. As the recording of data is done once per minute, the minimum temperature of the current data entry is the minimum value of the next 60 temperature values.

To fit the data structure for RNN, LSTM, and GRU models, sequences of data are created in step 3. A sequence contains all the features (Air Temperature, Dew Point, Relative Humidity, Wind Speed, Wind Direction). For every data entry, the current sequence is defined as sequence t, the sequence from one minute before is added to the current entry and defined as sequence $t-1$. Similarly, the sequence from two minutes before is added and defined as sequence $t-2$. Since the maximum sequence length of the experiments is 120, sequences from previous entries are added to the current entry from one minute before until 119 minutes before. Therefore, each entry includes 120 sequences from sequence $t-119$ to sequence t.

The timestamp column is a tool to help extract the data from the desired time period. Now,

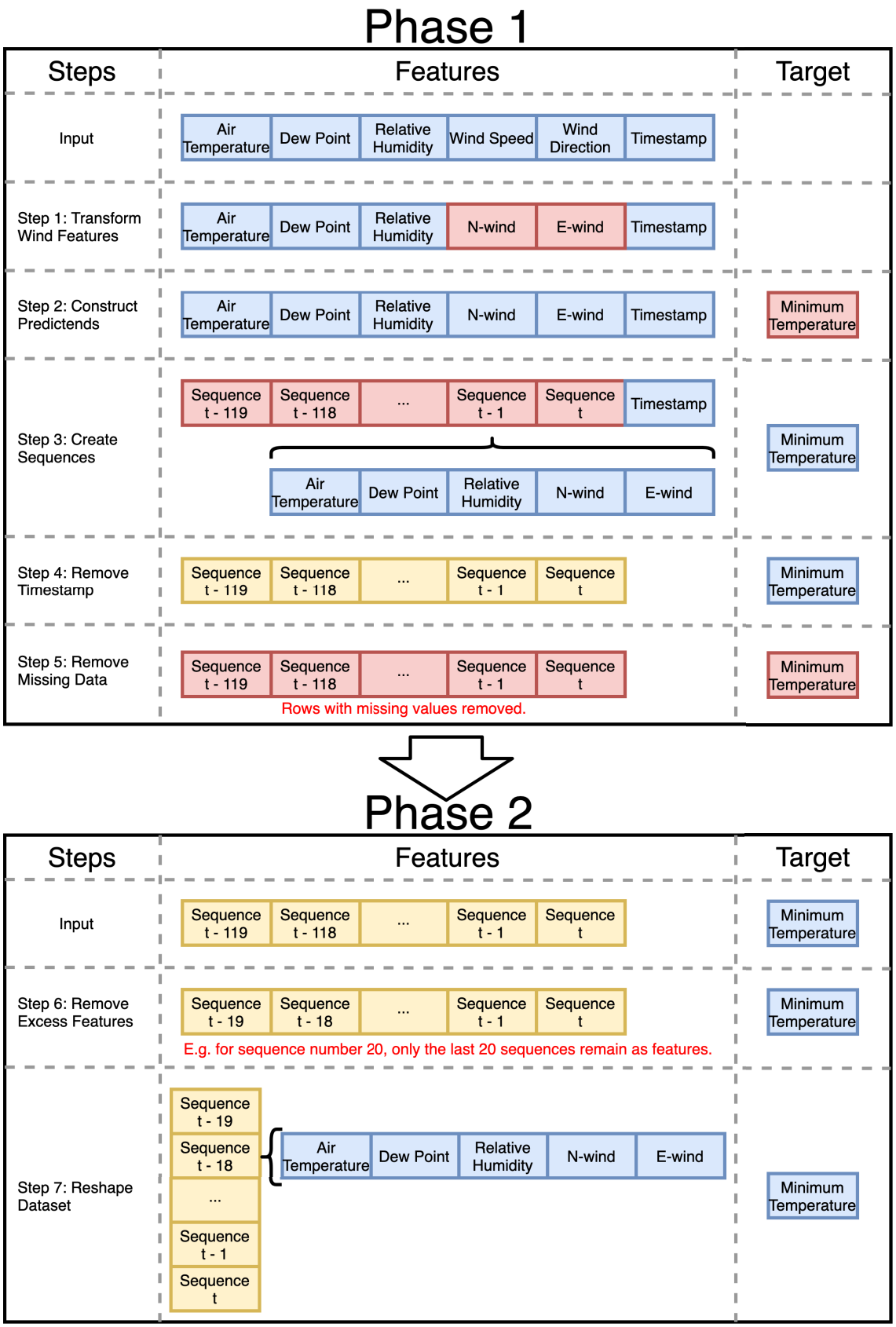


Figure 3.2: Data Preprocessing Steps.

as all of the features and sequences are generated, the timestamp column can be removed in step 4. Then, in the final step of phase 1, the listwise deletion data imputation technique is applied to eliminate data entries with empty features [266]. This also includes the removal of data sequences with missing features to preserve the time differences between observations. The final product of phase 1 is output to hard disk to be used in phase 2. For every weather station dataset, the steps in phase 1 are conducted for data in years 2016 and 2017.

In phase 2 of data preprocessing, the results from phase 1 are transformed to the required form for different models with different settings. After reading an output from phase 1, step 6 of data preprocessing removes excess data sequences. For ANN models, only one sequence is required. Therefore, sequences $t - 119$ to $t - 2$ should be removed. For RNN, LSTM, and GRU models, sequences are removed according to the sequence length of the target model. For example, with a desired sequence length of 20, sequences $t - 119$ to $t - 20$ are removed, leaving only 20 data sequences (Figure 3.2).

Step 7 of data preprocessing is only executed to prepare the data for RNNs and their variants. Every entry in the dataset is converted to a 2D structure. Each row of this 2D structure represents a sequence of a specific time. The rows are structured top to bottom from an earlier time to a more recent time. At this stage, the data can proceed to model construction.

3.2.3 Model Construction and Testing

In this subsection, information of model construction and testing is provided. The computation environment is firstly described, followed by the usage of datasets. Next, the model structures and hyperparameters are clarified. Finally, the two stages of model construction are revealed.

All models in this chapter are constructed on a desktop computer using an Intel i7-8700K 3.70 GHz processor, equipped with 32GB RAM. The graphics processing unit is a Nvidia RTX 2080 graphics card. The deep learning framework for model construction and testing is TensorFlow 2.3.0.

The data preprocessing processes generate two datasets from 2016 and 2017 for every weather station. In the experiment scene, models should be built in the current year and deployed in the next year. The year 2016 has been set as the “current” year and 2017 is the “next” year. Therefore, datasets from 2016 are used for model construction and datasets from 2017 are used for final testing.

It is assumed that during the process of model construction only the “current” year data is available. The datasets from year 2016 are split for model training, validation, and testing. For every dataset from different weather stations, 80% of the data are randomly allocated to the training dataset and the other 20% to the testing dataset. From the training dataset, a further split of 20% of the data forms the validation dataset. The training dataset, which contains most of the data, is used to fit the parameters of the models. During the training process, the validation dataset helps to tune the hyperparameters. After all training, hyperparameter tuning, and validation are completed,

test predictions are conducted by the model with the testing dataset. This provides an error metric of the model. However, as the model should be deployed in the "next" year, datasets from 2017 are used as extra and final testing datasets. In the experiments, the errors generated from the 2017 testing datasets are also compared with each other.

The model structure for all models is defined as a three-layer model. The first layer has five neurons. The second has seven and the output layer has one neuron. The first two layers change according to the target model. For example, if it is an ANN model, the cells in these two layers are ANN cells with Rectified Linear Unit (ReLU) activation functions. For RNN and its variants, the cells use the tanh activation functions with relevant cells to the models. Also, for the first layer of RNN, LSTM, and GRU models, the hidden state of the cells is output as sequential inputs of the second layer. The third layer only consists of a single linear cell to output the result.

Adam is the optimizer used during training. Learning rate, β_1, β_2, and ϵ, are the hyperparameters required for Adam [267]. In the experiments, learning rate is set to the "good default settings" as 0.001, β_1 as 0.9, β_2 as 0.999, and ϵ as 10^{-7}. Another hyperparameter, batch size is configured to 64. With these settings, all models are trained for 100 epochs with a Mean-Square Error (MSE) loss function.

Models are constructed in two groups. The groups are separated by different model structures and settings. In the first group, one ANN model is constructed for each of the 30 weather stations. In the second group, RNN, LSTM, and GRU models with different sequence lengths are trained for all 30 weather stations. There are 6 different sequence length settings (20, 40, 60, 80, 100, 120). Altogether, the second group outputs 540 models. Overall, there are 570 models constructed for this article. The usage of these models in experiments is explained in the next subsection.

3.2.4 Experiments

There are three experiments conducted to test the model error and performance of models. In these experiments, model error is measured in MSE. MSE is computed with $\sum_{i=1}^{D}(y_i - \hat{y}_i)^2$, where D is the number of cases in a sample, y_i is the observed result, and $\hat{y}_i$ is the predicted result. In the first experiment, the errors of different model types (ANN, RNN, LSTM, GRU) are compared with each other. ANN models from the first model group are the baseline of this experiment. For each weather station, there are eight conducted tests to measure the losses from the current year and next year data for the four model types. RNN models and their variants are tested with a sequence length of 120. As the result, the first experiment attempts to obtain an RNN-based model type with the lowest loss in the current year and next year settings. Then, in the second experiment, the effects of the sequence length of RNN, LSTM, and GRU models are revealed. Additional to the results obtained in the first experiment, 30 more tests are conducted for each weather station to obtain the results of the three RNN-based model types with five sequence number settings, and tested with the current year and next year testing datasets. Model errors of all RNN-based model types are compared with different sequence length settings against the ANN baseline. Similarly, in the final experiment, the training time and inference time of different sequence length settings

are compared against the ANN baseline.

3.3 Results and Discussions

To evaluate the performance of different model settings for next hour frost prediction models, the experiments defined in the above section are conducted. The first experiment compared the MSE between ANN and RNN-based model types to determine the RNN-based model type with the lowest loss. Then, in the second experiment, MSEs of RNN-based models with different sequence lengths are assessed to reveal the effect of sequence length on model losses. Finally, processing time related factors are also analyzed with different sequence lengths. This provides an overview of different models' real-time computation abilities.

3.3.1 Model Error

In this experiment, the model errors of RNN-based models with a sequence length of 120 are evaluated with ANN models. Figure 3.3 shows that when testing with testing datasets derived from the same year when the training datasets is collected, LSTM seems to perform with the best accuracy with the lowest MSE loss. LSTM is also the only RNN-based model type to exceed the accuracy of ANN models. This result is also confirmed with one-sided paired T-tests. From Table 3.1, LSTM is the only model type that the p-value is smaller than the 0.05 α value. This means the null hypothesis is rejected and ANN is likely to produce outputs with a greater loss than LSTM. LSTM models have the highest accuracy among other RNN-based models due to the extra gates to memorize sequence patterns [256].

An assumption made for the models of this chapter is that the models are constructed using data from "this" year and deployed in the "next" year. Therefore, models are also tested with testing datasets obtained one year after the training datasets. The results are significantly different compared to the results from the "current" year testing datasets (Figure 3.3). ANN shows the lowest MSE loss, which indicates the highest accuracy. On the other hand, LSTM models have the lowest accuracy. Table 3.2 demonstrates that LSTM and GRU models have p-values for one-sided paired T-tests less than the α threshold. Therefore, it is likely that LSTM and GRU models all have a significantly higher loss (less accurate) than ANN models.

As all models are trained with the current year data, LSTM models with more parameters and gates [268] fit closer to the current year testing datasets. On the other hand, RNN-based models are constructed through learning the sequence patterns [112, 256]. Thus, these models are sensitive to the change of sequence patterns. In [29, 30], the global climate change induces an increase of instability in weather patterns over time. As a result, the accuracy of RNN-based models deteriorates when tested with the next year testing datasets. Compared to the baseline, LSTM and GRU models with more parameters [268] tend to "overfit" more to the current year pattern and are vulnerable to the changed next year pattern. However, the exact extent of accuracy reduction is unknown as the change of climate patterns in the future is also unknown.

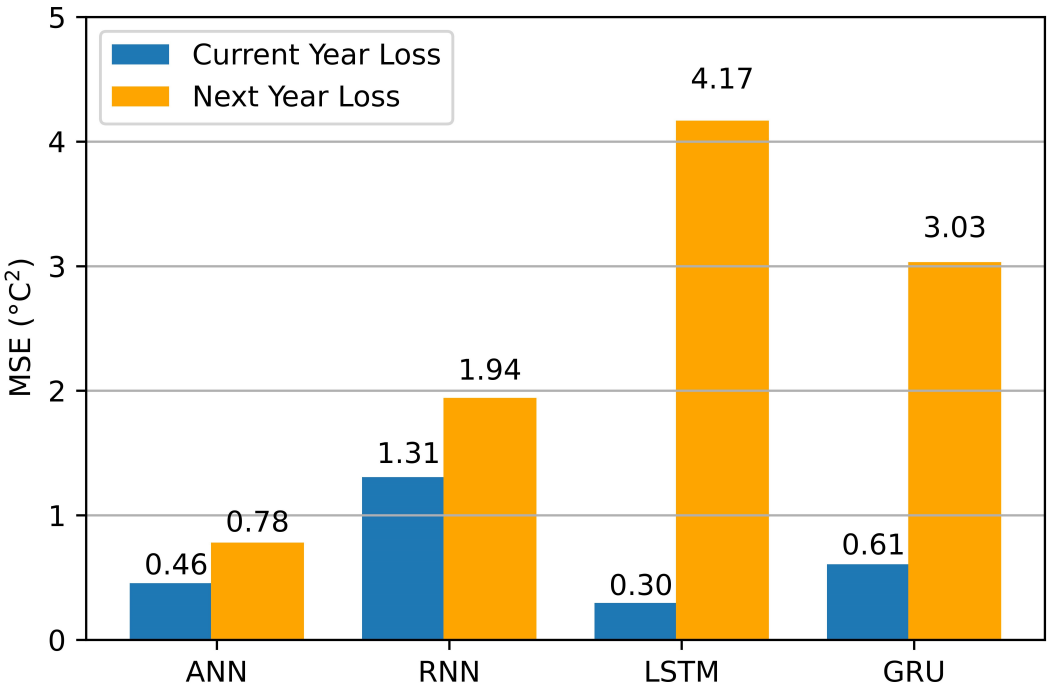


Figure 3.3: Average MSE Tested with Current and Next Year Datasets

Table 3.1: P-values of Comparing Average MSEs for Current Year Datasets

	RNN	**LSTM**	**GRU**
ANN	0.1544	6.1225e-12	0.2644

Table 3.2: P-values of Comparing Average MSEs for Next Year Datasets

	RNN	**LSTM**	**GRU**
ANN	0.1350	0.0432	0.0027

3.3.2 Effect of Sequence Length on Model Error

In this experiment, the effect of sequence length on model error for RNN-based models is inspected. Figure 3.4 shows the average MSEs of RNN-based models tested by current year datasets. Overall, the increase of sequence length does not reduce the average loss. Only LSTM shows a decreasing trend of losses with the increase of sequence length. However, this change is not very significant. When compared to ANN, LSTM and GRU with some settings (sequence length=20, 40, 80, 100) seem to have a smaller loss than ANN (0.4550). This is confirmed by the one-sided paired T-tests results on Table 3.3. Only p-values of LSTM, and GRU models with the sequence lengths of 20, 40, 80, and 100 are smaller than the α. This reveals that when testing with the current year datasets. This demonstrates the higher likelihood that LSTM and GRU (sequence length=20, 40, 80, 100) models are more accurate than ANN models when tested with current year testing datasets.

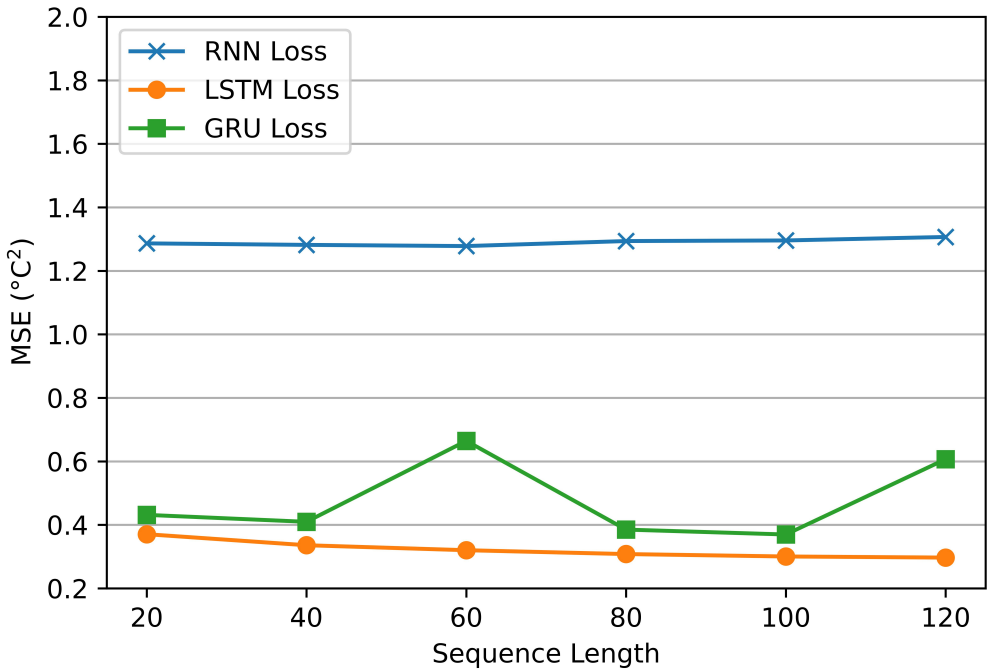


Figure 3.4: Average MSE Tested with Current Year Datasets for Different Sequence Lengths

Figure 3.5 is the average MSEs for RNN-based models with different sequence lengths tested by the next year testing datasets. Similar to the reverse of results in Experiment 1, all RNN-based models have a higher loss than the ANN MSE (0.7813 $^oC^2$). Table 3.4 is the p-values obtained from one-sided paired T-tests with an alternative hypothesis that each tested model has a greater MSE than the ANN baseline. The alternative hypothesis is in favor of LSTM and GRU models as their p-values are less than α. This means LSTM and GRU models are likely to perform with higher errors than ANN models in the next year. Also, as explained in Experiment 1, the change of climate patterns in the future is unknown. This could be the reason of the additional noise in

Table 3.3: P-values of Comparing Average MSEs between ANN and RNN-based Models with Different Model Sequence Lengths (Current Year)

Model Type / Sequence Length	RNN	LSTM	GRU
20	0.1603	3.4979e-10	0.0032
40	0.1617	6.9911e-11	1.3249e-5
60	0.1628	1.0544e-10	0.1591
80	0.1581	3.0000e-11	4.2542e-7
100	0.1576	2.1009e-11	4.2930e-7
120	0.1544	6.1225e-12	0.2644

Figure 3.5, compared to Figure 3.4.

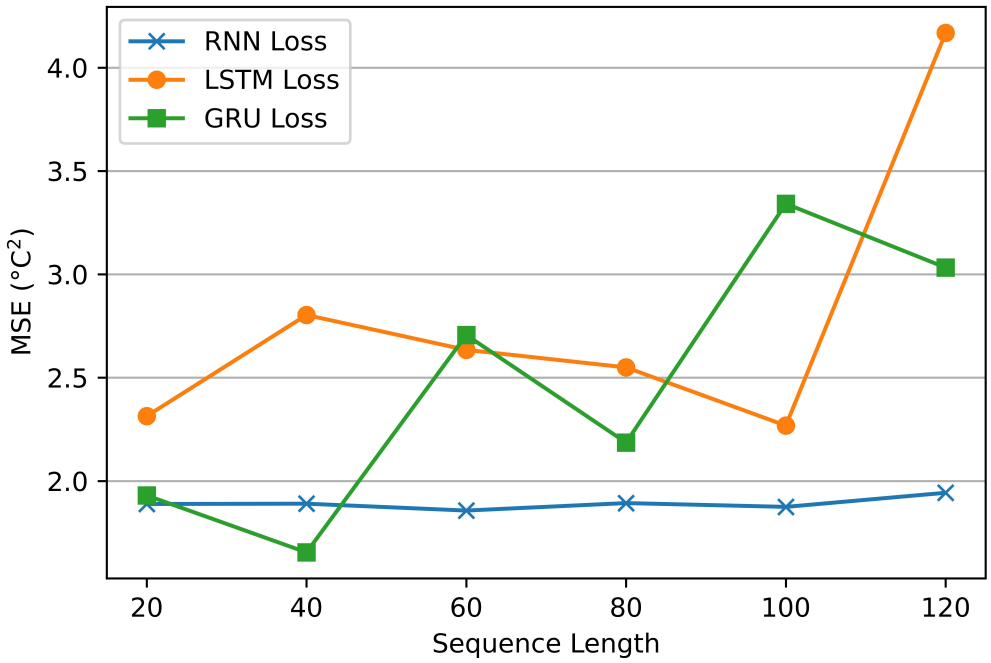


Figure 3.5: Average MSE Tested with Next Year Datasets for Different Sequence Lengths

3.3.3 Effect of Sequence Length on Processing Time

In the final experiment, the training time per epoch and inference time per input are tested and compared between ANN models and RNN-based models with different sequence lengths. Figure 3.6 demonstrates that the training time per epoch for RNN, LSTM, and GRU models increases as the sequence length increases. This is due to the additional parameters for training as the sequence

Table 3.4: P-values of Comparing Average MSEs between ANN and RNN-based Models with Different Model Sequence Lengths (Next Year)

Model Type / Sequence Length	RNN	LSTM	GRU
20	0.1466	0.0013	0.0109
40	0.1463	0.0040	0.0249
60	0.1538	0.0028	0.0156
80	0.1457	0.0072	0.0089
100	0.1497	0.0048	0.0294
120	0.1350	0.0433	0.0028

length increases. LSTM and GRU models have overlapping curves of training times. RNN training times are greater than the prior two models. This is because, in TensorFlow 2.3.0, only LSTM and GRU model training processes are optimized by cuDNN for faster training speeds [269, 270]. However, whether optimized or not, compared to the ANN training time of 1.070 seconds, as the p-values are smaller than the 0.05 α, Table 3.5 indicates that the training time of ANN models per epoch is more likely to be smaller than all settings of RNN-based models.

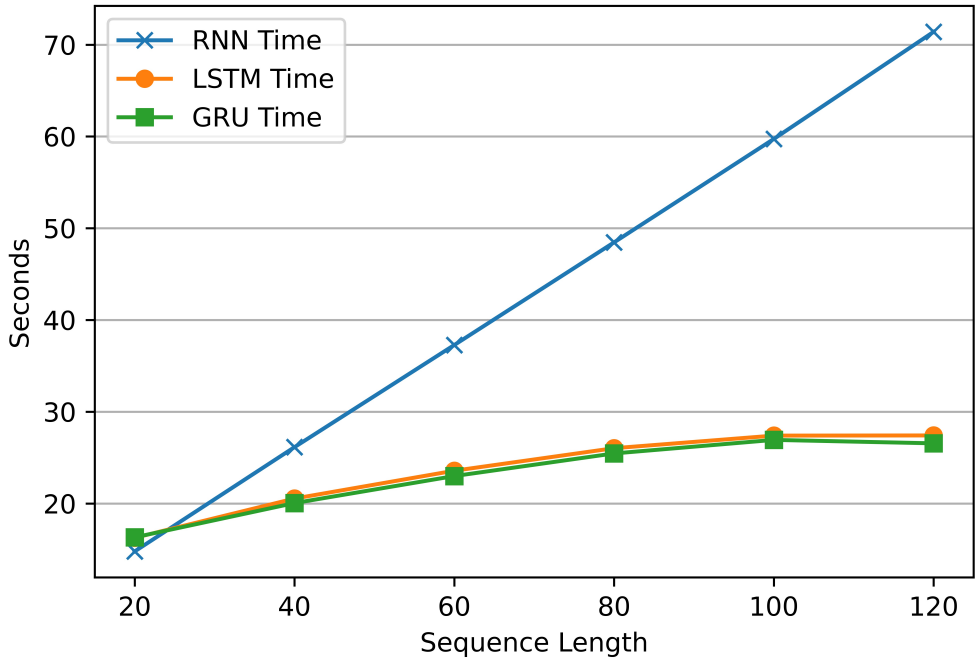


Figure 3.6: Average Training Time per Epoch for Different Sequence Lengths

Figure 3.7 demonstrates the average inference time per input for RNN-based models with different sequence lengths. There is a trend of increase in inference time, along with the increase

Table 3.5: P-values of Comparing Average Training Time per Epoch between ANN and RNN-based Models with Different Model Sequence Lengths

Model Type / Sequence Length	RNN	LSTM	GRU
20	0	0	0
40	0	0	0
60	0	0	0
80	0	0	0
100	0	0	0
120	0	0	0

of sequence length. Similar to training time, a higher sequence length of RNN-based models implies a larger input sequence and a model structure with more parameters. Thus, the inference time increases with the sequence length. Also, the inference time for all RNN-based settings is significantly larger than ANN inference time (4.5214e-4 seconds). This statement is supported with the one-sided paired T-test results as all p-values are less than the α (Table 3.6).

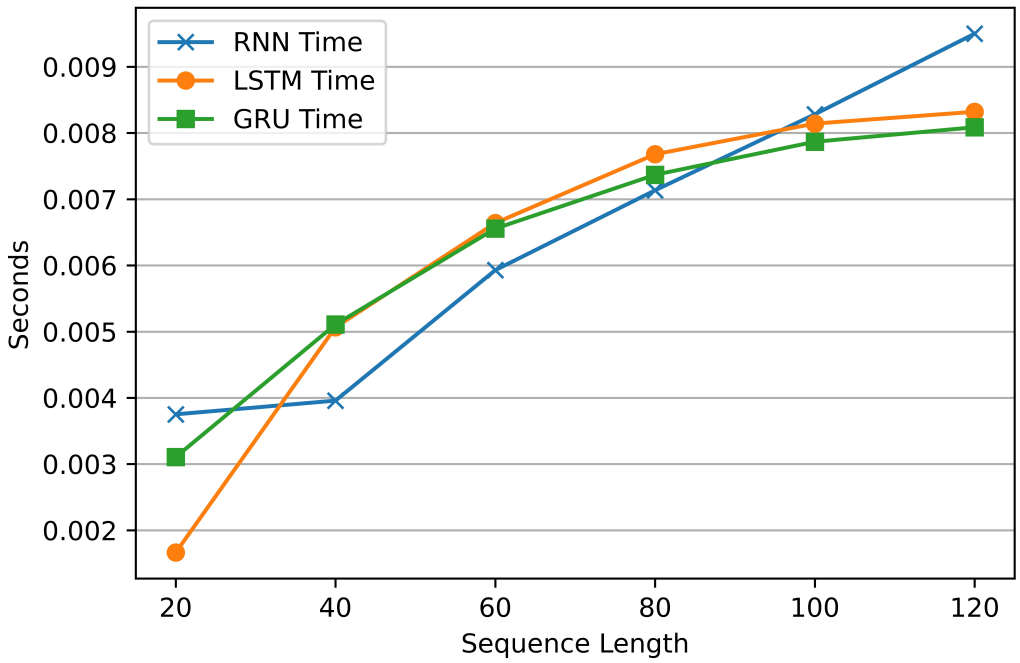


Figure 3.7: Average Inference Time per Input for Different Sequence Lengths

Table 3.6: P-values of Comparing Average Inference Time per Input between ANN and RNN-based Models with Different Model Sequence Lengths

Model Type / Sequence Length	RNN	LSTM	GRU
20	0	1.0941e-17	0
40	0	0	0
60	0	0	0
80	0	0	0
100	0	0	0
120	0	0	0

3.3.4 Limitations and Open Challenges

Experiment 1 shows that RNN-based models have lower model errors than ANN models when tested with current year datasets. LSTM models have the lowest errors and highest accuracy. However, RNN-based models' accuracy declines and is exceeded by ANN when tested with next year datasets. This decline is likely to be caused by climate pattern change over time [29, 30, 112, 256]. Also, the models in this chapter are constructed with one year of data. From [271], RNN models often require data from more years to fully learn the seasonality patterns. However, this would increase the dependency on historical data, which is a limitation mentioned in the next subsection. With both sources of accuracy deterioration, RNN-based models are only suitable for a model deployed in the short term. In Experiment 2, RNN-based models do not present a significant reduction in model error as the sequence length increases. Therefore, considering the results from Experiment 3, RNN-based models with a short sequence can be deployed for short times to provide predictions with higher accuracy and performance. On the other hand, ANN models can be deployed over longer time spans without much accuracy deterioration. Both training and inference time is significantly lower than RNN-based models. Overall, ANNs might still be more suitable than RNN-based models for frost prediction in a long-term scenario with minimal system maintenance and update because of their higher accuracy and performance over the long term. After constructing the above different frost prediction models and analyzing their performance, the limitations presented in this subsection are discovered. Limitations create new challenges and lead towards future directions to frost prediction research.

3.3.4.1 Model accuracy requirements are not specified

Different plants have different frost tolerance and sensitivity [260]. Therefore, model accuracy requirements may vary between different plants. There are many studies on plant frost tolerance. However, the sensitivity of plants to individual frost factors is not fully revealed [272]. As a future direction, the sensitivity to different frost factors such as temperature, humidity, dew point,

cloud coverage, solar radiation and wind speed should be tested with high precision for individual species of plants in controlled environments. With enough plant species studied, a model accuracy threshold can be set for frost prediction models.

3.3.4.2 Lack of standard datasets

The models and experiment results are obtained from a public dataset that represents local climate conditions of our study sites in Australia. Therefore, the accuracy of the models at other sites with different climate patterns is questionable. To the best of our knowledge, prior research has been based upon locally obtained private datasets [207, 213, 216–219]. As a result, model results could be biased. There is a need of a standard dataset for frost prediction models that includes data entries from different locations.

3.3.4.3 The RNN input data format affects the system energy efficiency

The RNN-based models in this chapter require a sequence of climate data as model inputs. Each sequence of data contains sensor readings collected every minute over a long time span. This places a restriction on node duty cycles. Sensor readings must be collected per minute to satisfy the model input. For common agricultural IoT systems, sensor readings are transmitted through the radio from a node to a central processing node for further processing and data analytics [273]. In [274], radio power consumption increases as the time interval of radio transmission reports decreases. Even radio transmissions between 10-minute periods consume a significant amount of the system energy [274]. Therefore, if sensor readings are reported every minute, higher energy consumption will be placed on the whole system. This issue could be mitigated by aggregating a few minute-wise sensor readings into one radio transmission. Also, inference on edge with edge computing can also reduce the number of radio transmissions. A model can be deployed on edge devices and only transmit the predicted outcome when it triggers a preset condition. However, even there are mitigation plans, RNN-based models still limit the design of IoT systems with potential high energy impact. On the other hand, each ANN model inference only requires a single set of climate data as input. ANN-based frost prediction models can be applied to systems with different time intervals to obtain sensor data. Radio transmission intervals can be adjusted according to system requirements. Therefore, ANN models may still be more suitable for frost prediction systems.

3.3.4.4 Models depend on previous year data

All models constructed for this chapter depend on previous years of data. Therefore, model accuracy is dependent on the quality of historical data. This is a limitation on all machine learning models [94]. As more recent researches are site-specific, sites without any record of historical climate data require an IoT data collection system to be deployed for data collection. To ensure similar performance to the results of this chapter, the system must be deployed at least a year

prior to produce a prediction model and provide frost prediction services. This increases the deployment time of the system. A possible solution is to explore the performance of models based on previous months of data. This solution could substantially reduce development time. However, it cannot eliminate this excess system development time for data collection. Methods eliminating the requirement of on-site historical data need to be developed. Another possible direction is to explore the generalization or transfer of models to similar locations.

3.3.4.5 Lack of stopping conditions

The frost prediction models constructed in this chapter only predict the start of a frost event with real-time sensor readings. This prediction could be the trigger of a frost protection mechanism. However, there are limited mentions of predicting the end of a frost event to switch off any protection mechanisms. As most frost protection systems rely only on a single sensor node [34], activation of a nearby protection mechanism might affect the sensor readings and contaminate the prediction outcomes of frost prediction models. Therefore, future prediction models could be developed to eliminate the effect of frost protection mechanisms. As a possible benefit, frost protection mechanisms could be switched off earlier to reduce the operational cost.

3.4 Summary

As a response to **Research Question 4**, the primary aim of this chapter is to increase the prediction frequency from once per 12–24 hours for the next day or night events to minute-wise predictions for the next hour events. RNN-based models are selected to learn the sequence pattern of historical data. ANN models are used as a baseline. Datasets from weather stations in the NSW and ACT areas of Australia are obtained. These datasets are recorded during the years 2016 and 2017. With these datasets, it is assumed that our models are built during the year 2016 (current year) and deployed in year 2017 (next year). Therefore, datasets from 2016 are used for model construction and preliminary testing. Datasets from 2017 are used for final testing. After constructing the models, there are three experiments testing the model errors, also the effect of sequence lengths on errors and processing time for RNN-based models. The errors of models is tested with both the current and next year datasets. LSTM seems to have the highest accuracy when tested with the current year testing datasets. However, the accuracy for all RNN-based models reduces when tested with the next year testing datasets. ANN models have the highest accuracy with the next year testing datasets. When testing RNN-based models with different sequence lengths, it seems that sequence lengths can not affect the accuracy of models significantly. However, training and inference time increases with the sequence length. Therefore, RNN-based models should be used for short-term deployments with a shorter sequence length to ensure accuracy and performance. On the other hand, ANN models demonstrate the lowest error when tested with next year datasets. Also, the training and inference speeds of ANN models are faster than RNN-based models. Therefore, in the long term, ANN models are more suitable than RNN-based models due to better accuracy and

performance.

There are limitations determined. Firstly, the model accuracy requirements are not specified in this chapter due to the lack of studies on precise frost sensitivities to individual frost factors. Secondly, the current model and most previous models are constructed with local data. The lack of standard datasets limits unbiased comparisons between different models. Thirdly, the RNN input data format affects the system energy efficiency. The limitations on energy efficiency could be avoided by the application of edge computing or message aggregation. On the other hand, ANN models do not have such limitations. The next limitation is related to the construction of models. All of these models depend on previous year data. For sites without data records, this limitation induces extra development and deployment time. Therefore, models that decouple from historical data could be developed to eliminate this restriction completely. The final limitation is that prediction models lack stopping conditions. As most systems read climate data from one sensor node, the activation of protection gear could contaminate the input data. Thus, it contaminates the prediction outcomes. The last two limitations are addressed in the next chapters of this thesis. The limitation of historical data dependence has also been mentioned in Chapter 2. This limitation is handled in the next Chapter (Chapter 4) along with the issue of low fault tolerance (**Research Question 5**). Combining the results of this chapter and Chapter 4, a model is developed to remove the effect of data contamination and provide a model-based automatic stopping mechanism to reduce operational costs of frost protection systems.

4

Intelligent Spatial Interpolation-based Frost Prediction Methodology using Artificial Neural Networks with Limited Local Data

4.1 Introduction

From the studies of Chapters 2 [34] and 3 [257], recent frost prediction models require historical data from a specific site to predict future frost events. However, this requirement of on-site historical data poses a restriction for model construction in new sites without historical data. Long periods of time are needed for data collection at these new sites to construct and deploy the machine learning models. After the models are constructed, local sensor motes are needed to feed live data to the prediction models. This is another restriction of recent methods. To overcome these restrictions, this chapter proposes a frost prediction method based on spatial interpolation techniques, aiming to predict frost for a site without any on-site historical data or sensors. This chapter also forms part of the response to **Research Question 5** and loosens the frost prediction data dependency from the local data source as the only option to different external data sources as multiple candidates. Spatial interpolation is leveraged to support the proposed system.

Spatial interpolation includes methods that generate or predict spatially continuous data from a few regional sample points [275]. The authors of [276] compared different spatial interpolation methods for monthly air temperatures at Mt. Kilimanjaro, Tanzania. They found that model averaging neural networks and ANNs are ranked fourth and fifth in accuracy, respectively. ANNs are also the most accurate model type within recent frost prediction techniques. Therefore, in this

chapter, the proposed method uses ANN as the base model and baseline.

The proposed method is an ensemble learning method that utilizes the existing historical data from 75 weather stations across NSW and ACT in Australia. Several weak predictors are trained. Each weak predictor is created using the climate features from one specific station to predict the next hour minimum temperatures of other weather stations. Location features from both stations are also included. The location features include geographical location, elevation from the Digital Elevation Model (DEM), and Normalized Difference Vegetation Index (NDVI), indicating the amount of green vegetation [277]. These location features are also used to aggregate the results of the weak predictors. Unlike the climate features, the location features are near perpetual (geographical location, DEM) or change infrequently (NDVI). The correlation between NDVI and temperature is discovered in different works. Strong positive correlations are revealed between NDVI and minimum temperature [278, 279]. Therefore, NDVI is used as a hint for the minimum temperature. The purpose of adopting multiple weak predictors is to provide fault tolerance to external data sources. If a data source is unavailable or malfunctioning, the system can still operate with the other available data sources or weak predictors.

4.1.1 Related Work

Recent frost prediction methods can be divided into "classification methods" and "regression methods" [34]. The classification methods predict the probability of frost occurrence in the future. The regression methods predict the future minimum temperature. As different crops have different resistance to frost [260], this chapter focuses on regression methods of frost prediction. Along with the regression methods, different triggering temperatures for frost can be applied as a general solution for different types of crops and plants.

The majority of recent frost prediction regression methods [207, 213, 216–219] predict the minimum temperature of the next 12–24 hours. The prediction models include, linear regression [216, 219], random forest [207], and ANNs [213, 217, 218]. The ANNs have the highest accuracy among the other model types [34].

All these methods [207,213,216–219] depend on local/on-site historical climate data for model training, validation, testing and future operations. This first generates extra development and deployment time to collect local climate datasets. Then, a live sensor system is required to transmit data to the prediction models. The proposed method in this chapter aims to remove the model dependency of on-site data and sensors. Models can be built from existing historical data from other weather stations, previously surveyed DEM, and NDVI data from satellites. During operation time, the model input sources are also climate data from other weather stations, previously surveyed DEM, and NDVI data from satellites. The proposed method does not require on-site data or sensors. The major contributions of this chapter are:

1. Proposing a spatial interpolation-based frost prediction method.

2. Eliminating the on-site data/sensor requirement for frost regression prediction methods.

3. Exploring the performance of an off-site frost prediction method.

The rest of the chapter is organized as follows. Section 4.2 presents the methodology. This includes the descriptions of data sources, data preprocessing methods, model types, and experiment settings. The experiments compare the performance of the proposed method with the baseline method derived from on-site climate datasets. Then, Section 4.3 shows the experiment results along with the discussions on limitations. Finally, the chapter is summarized in Section 4.4.

4.2 Methodology

In this section, the data processing and experiment procedures are presented. To start all experiments, multiple datasets are obtained from different data sources. These datasets and their sources are described. Then, the data preprocessing steps are demonstrated, followed by the model structure of the prediction models. Finally, this section explains the experiment settings.

4.2.1 Data Sources

The data sources can be grouped into four categories. These categories are climate data, DEM data, NDVI data and state boundary data. The climate datasets are obtained from 75 different weather stations across NSW and ACT. The raw datasets can be obtained from the Australian BOM website [261]. In the climate data category, there are two groups of datasets. The first group consists of datasets from the year 2017. These datasets are used for model training, validation, and preliminary testing. The second group is datasets from June, July, and August (winter [280]) of 2018. These datasets are used to test the model performance in the winter one year after the models are constructed.

The second data source category includes a DEM dataset of the study area. The DEM dataset contains 1 second (about 30 m) resolution elevation data of Australia. The NSW and ACT portion of the DEM dataset is extracted to be used as features during model construction, validation, and testing. The raw DEM dataset is hosted by Geoscience Australia [281].

The raw NDVI datasets are collected from the Land Processes Distributed Active Archive Center of NASA [282]. The data resolution is 250 m. Similar to the climate datasets, NDVI datasets are from the year 2017 and the winter months of 2018. The datasets from 2017 are also used for model training, validation, and preliminary testing. The NDVI datasets from 2018 act as features during the final testing phase to evaluate the model performance in the winter one year after model construction.

Boundary datasets form the final data category in this chapter. There are two separate data files that contain boundary information of NSW and ACT. The area coverage is defined by coordinates of multiple polygon vertices. The boundary datasets are not directly involved in the process of model training, validation, or testing. However, the boundary datasets are required when extracting

the NSW and ACT portion of the DEM and NDVI datasets. The boundary datasets of NSW and ACT can be obtained from [283] and [284], respectively.

4.2.2 Data Preprocessing

Data preprocessing procedures are described according to the data categories defined in the above subsection. Climate data preprocessing is explained first, followed by DEM data preprocessing and NDVI data preprocessing. The boundary data is used during the preprocessing of DEM and NDVI datasets. However, as boundary data is not included in the processes of model training, validation, and testing, the usage of boundary data is only presented with other data categories. After the preprocessing of the four data categories, the datasets are merged to be fit as training data of prediction models. This merging process is also explained.

Each climate data file stores climate data from a different weather station. The temperature, dew point, relative humidity, wind speed, and wind direction fields are extracted from each of these stations. Wind speed and direction are converted to *N-wind* and *E-wind*, as the north and east components of the wind. This conversion is done by reversing the wind direction (met) to the wind blowing direction (deg) (Equation 3.1) [264]. Then, calculating the magnitude of the eastward (v_E) and northward (v_N) wind components with the wind speed (v) and wind blowing direction (deg) (Equation 3.2). After computing the wind components, the next hour minimum temperature is calculated as the training targets of the models. For each time step at each weather station, the target is obtained from the minimum temperature of the next 60 time steps.

Since the spatial interpolation models are built with five-fold validation, the weather stations are randomly divided into five folds. For each model fold, one different fold of the weather stations (15 stations) is used as testing data sources and the rest of the weather stations are used as training data sources. Figure 4.1 summarizes the above steps of five-fold validation and Figure 4.2 shows the distribution of weather station folds. Detailed coordinates of the fold stations are recorded in Appendix A.

The DEM dataset is resampled to a grid with a cell size of $0.01^\circ \times 0.01^\circ$ (approximately 1.11 km $\times$ 1.11 km). Then, the NSW and ACT parts of the map are extracted according to the boundary datasets. The DEM readings of Figure 4.2 is the result of DEM preprocessing. The NDVI datasets are preprocessed with the same procedures as the DEM dataset. These datasets are also resampled to a $0.01^\circ \times 0.01^\circ$ grid. Then, data is extracted within NSW and ACT boundaries.

Table 4.1 shows five feature groups. The training data required by all models in this chapter can be represented by assembling some of the five feature groups. There are two data input formats that correspond to the two model types. The first model type is the baseline model. The baseline model represents the previous models relying on local sensors and data. It is used as a benchmark to reveal any gaps between the previous method and the proposed method (off-site prediction). Therefore, the source station climate features are required as the input features of the baseline models, and the next hour minimum temperature of the source station is used as the model target or predictand.

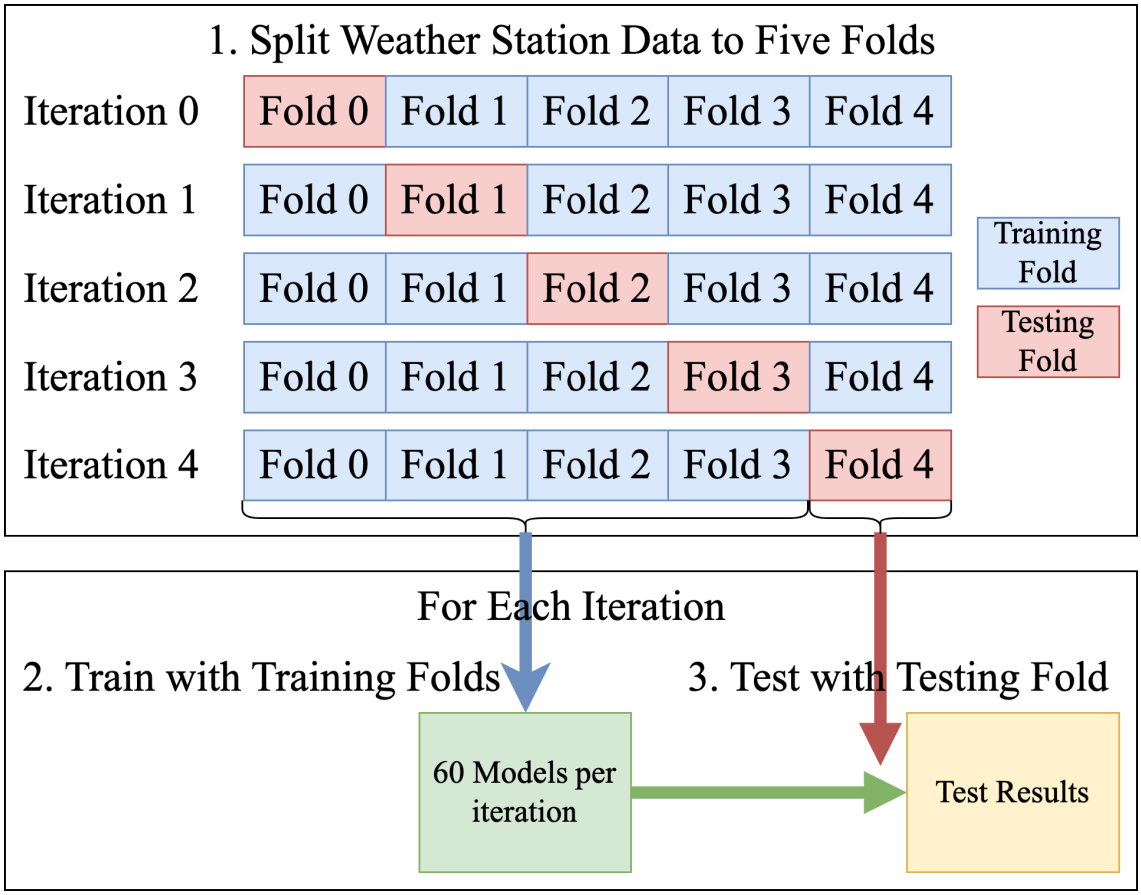


Figure 4.1: Training and Testing with Five-fold Validation.

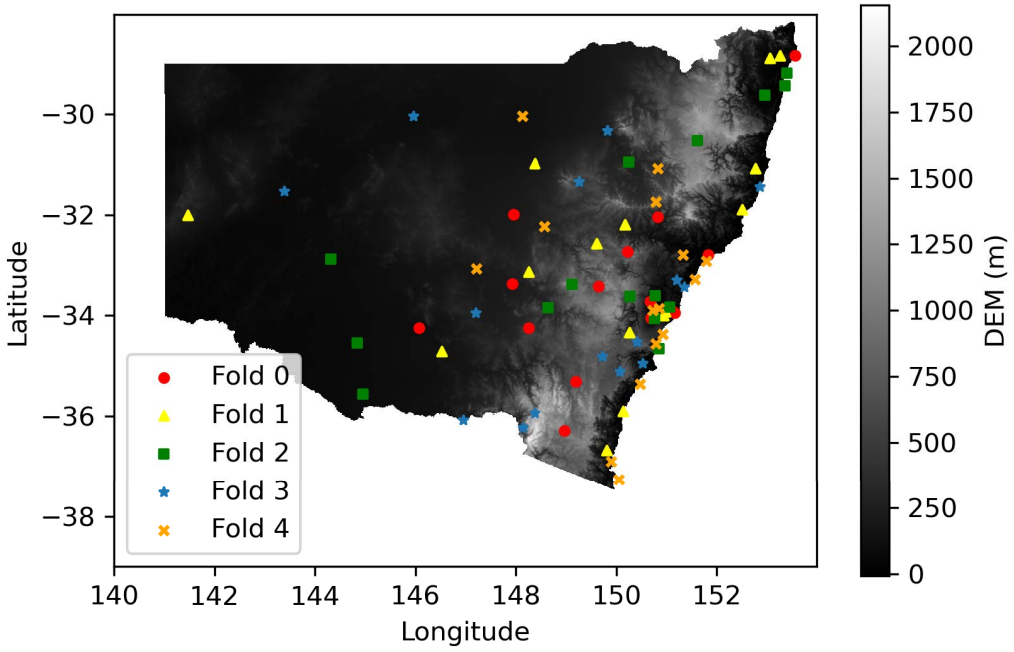


Figure 4.2: Distribution of Station Folds on NSW and ACT DEM Map.

The second model type provides off-site frost predictions. For each model prediction, the models utilize climate data from a source station and predict the next hour minimum temperatures for a target station in another locations. The source station attributes and target station attributes are acquired as the features to compute the differences between stations (Location, DEM, NDVI). Also, the source station climate features are input as climate references. Finally, the spatial model predictand is the minimum temperature values from the target station.

Table 4.1: Features and Predictands of Preprocessed Datasets

Feature Groups	**Features**
Source Station Attributes	Longitude, Latitude, DEM, NDVI
Target Station Attributes	Longitude, Latitude, DEM, NDVI
Source Station Climate Features	Temperature, Dew Point, Relative Humidity, N-wind, E-wind
Baseline Predictand	Source Station Minimum Temperature
Spatial Model Predictand	Target Station Minimum Temperature

4.2.3 Model Types

This chapter involves two major model types. They are the baseline models (benchmark) and spatial interpolation-based models (proposed). Baseline models are ANN models that harness data from sensors at a particular site and predict frost on the site. Every model is designed to predict frost for one different location. These models represent recent frost prediction models that are built from on-site historical climate datasets.

Spatial interpolation-based models are also ANN models. Unlike the baseline models, the spatial interpolation-based models do not need on-site sensor data from the target location. These models use off-site climate data from weather stations of other locations to predict any frost of the target location. In this chapter, there are multiple spatial interpolation-based models. Each model in each station fold is constructed based on the climate data of a different source station, and the target stations of that model are the testing stations of the fold. To increase the accuracy of predictions, the results within a fold are aggregated. Two results aggregation methods are tested. The first method is averaging all results within each fold. The second method is to compute a weighted average based on the difference between each source station and the target location/station. This difference is calculated by geographical coordinates, DEM, and NDVI.

4.2.3.1 Weighted Average Layer

The weighted average layer is applied after obtaining the results of multiple spatial interpolation-based models in a particular fold. A weight is calculated for every station with an available prediction result. An intermediate weight for the *i*th source station (W_i) is first calculated by Equation 4.1. g_i, d_i, and n_i are normalized geographical distance, DEM difference, and NDVI difference between the source station and the target location. The geographical distance is computed as the two-dimensional Euclidean distance d_2 (Equation 4.2). x_1, y_1 and x_2, y_2 are the longitude, latitude of two locations in Equation 4.2. The DEM difference is computed as the one-dimensional Euclidean distance d_1. In Equation 4.3, v_1 and v_2 are the DEM of two locations. NDVI difference is also calculated as the one-dimensional Euclidean distance by replacing v_1 and v_2 in 4.3 to the NDVI of two locations. To compute g_i, d_i, and n_i, the geological distance, DEM difference, and NDVI difference are normalized in Equation 4.4, with d_i as the input distance/difference, d_{min} is the minimum distance/difference and d_{max} as the maximum. a, b, and c provide adjustable importance of the geological distance, DEM difference, and NDVI difference to the station weight. Currently, the adjustable weights (Table 4.2) are generated by Pearson Correlation [285] between each of the distance/differences and model errors computed by absolute error. After the intermediate weights for all stations are obtained, these intermediate weight for each station is divided by the sum of the intermediate weights W_{sum} to obtain the final weight F_i for each station (Equation 4.5).

$$W_i = \frac{1}{ag_i + bd_i + cn_i} \tag{4.1}$$

$$d_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.2}$$

$$d_1 = |v_1 - v_2| \tag{4.3}$$

$$\bar{d} = \frac{d_i - d_{min}}{d_{max} - d_{min}} \tag{4.4}$$

$$F_i = \frac{W_i}{W_{sum}} \tag{4.5}$$

Table 4.2: Adjustable Weights for Different Folds

Fold Number	**Geo Weight** (a)	**DEM Weight** (b)	**NDVI Weight** (c)
0	0.1629	0.0132	0.0290
1	0.1768	0.0205	0.0238
2	0.1612	0.0222	0.0177
3	0.1804	0.0114	0.0269
4	0.1601	0.0110	0.0260

4.2.4 Experiments

To compare the results of the off-site prediction models with the baseline, three experiments are conducted. The aim of the first experiment is to compare the prediction results of different folds and determine if bias exists due to different station training data. Three sets of raster maps of NSW and ACT are generated from the datasets acquired in the year 2017. These raster maps are computed from the spatial interpolation-based models. Each map of the first set is created using the results from one different weather station. The second set is created by averaging the first map set per fold. After that, the final map set is generated using the weighted average layer.

The second experiment compares the accuracy between the three model types (baseline, averaging, weighted averaging). The testing datasets are from 2017. The baseline models use parts of the datasets not involved in the training process (20% of data for each station from 2017). For spatial interpolation-based models, 15 weather stations per fold act as testing datasets (Appendix A).

In the final experiment, datasets from the winter months of 2018 are used to compare the spatial interpolation-based models with different numbers of available stations. Model accuracy is compared against the baseline models. After that, the percentages of captured events below zero degrees [199] are also evaluated. An event is determined as a time step, where the temperature is

below zero degrees. An event is captured when the prediction algorithm predicts that the temperature is below zero degrees at the time step of the event. Finally, the results of the proposed methods are compared with the results of Inverse Distance Weighting (IDW) and Ordinary Kriging (OK). IDW and OK are the two most commonly compared spatial interpolation methods [275].

4.3 Results and Discussions

4.3.1 Effect of Different Fold Training Datasets

In this experiment, the differences in the results generated by models trained and tested with different datasets are revealed. As examples, one test station is randomly chosen from each of the weather station folds. For each of the stations, four raster maps generated from models trained with climate data from the station are compared from the other four folds. The fives stations are stations 63291, 66137, 58212, 72160, and 67119, from folds 0, 1, 2, 3, and 4, respectively.

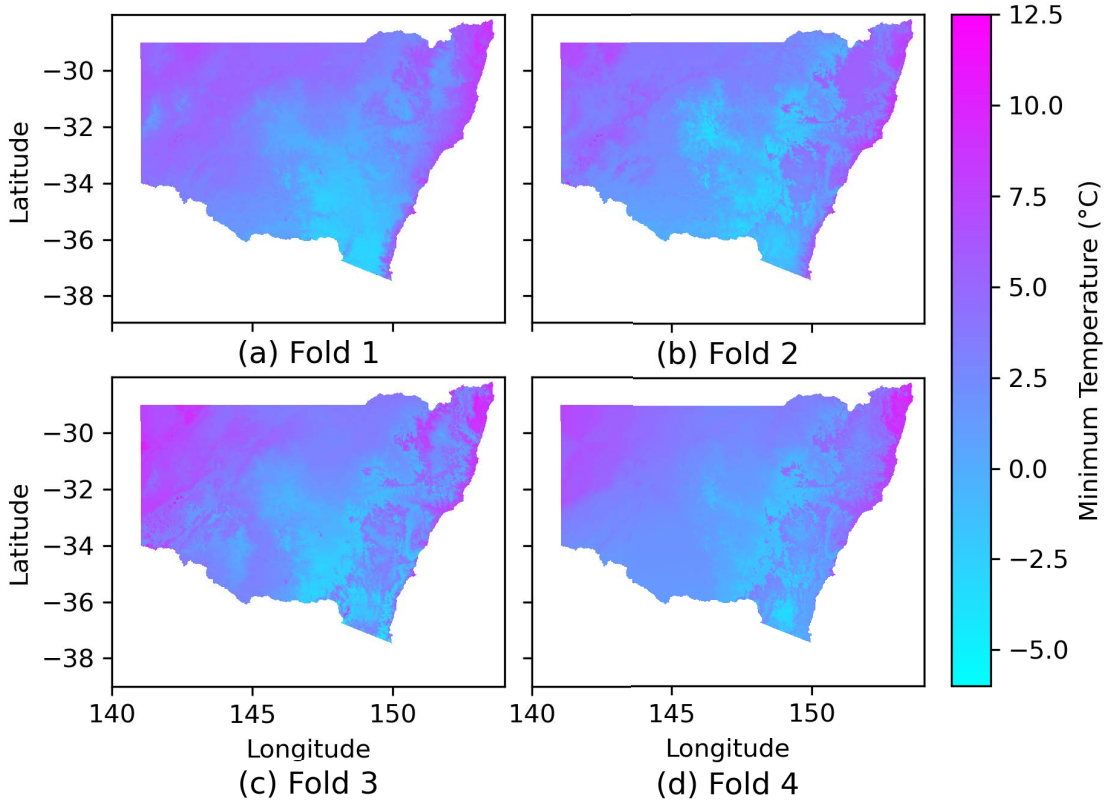


Figure 4.3: Raster Maps from Models Using Weather Station 63291 as Climate Data Source Trained with the Datasets from Folds 1–4.

Figure 4.3 shows the raster maps generated from individual models of station 63291 constructed by the training data (excluding the data from testing stations in each test fold) in fold 1–4. From a visual perspective, the raster maps of the four folds are similar in major features. For example, all maps present heat spots in the northeastern and northwestern regions. Also, all maps demonstrate the cold spot near the center of the maps. However, the shapes of these features are different between each fold. Folds 2 and 3 show more details than the other two folds. The maps seem to be significantly affected by the available training datasets for different folds. The result of T-tests also supports the hypothesis that raster map results are affected by the different training datasets in each fold. Paired T-tests are conducted for all possible fold raster map result pairs. All of the P-values are zero (smaller than 0.05). This rejects the null hypothesis and favors the alternative hypothesis that the raster maps are different from each other. Raster maps from stations 66137, 58212, 72160, 67119 draw similar conclusions. The raster maps of these stations are placed in Appendix B and the P-values are all zero.

Raster maps (Figure 4.4) are also generated for the averaged result of each folds. Compared to the maps generated from individual models, maps of averaging models per fold are more visually similar. However, The results of paired T-tests are similar to the individual models. All of the P-values are zero (smaller than 0.05). This suggests strong evidence against the null hypothesis. There are significant differences between these averaged maps from different folds. Absolute error maps are generated to visualize the difference between averaged results of different folds (Figure 4.5). These maps are generated by computing the absolute difference between corresponding points from two maps of different folds. These differences also exist between weighted averaged maps.

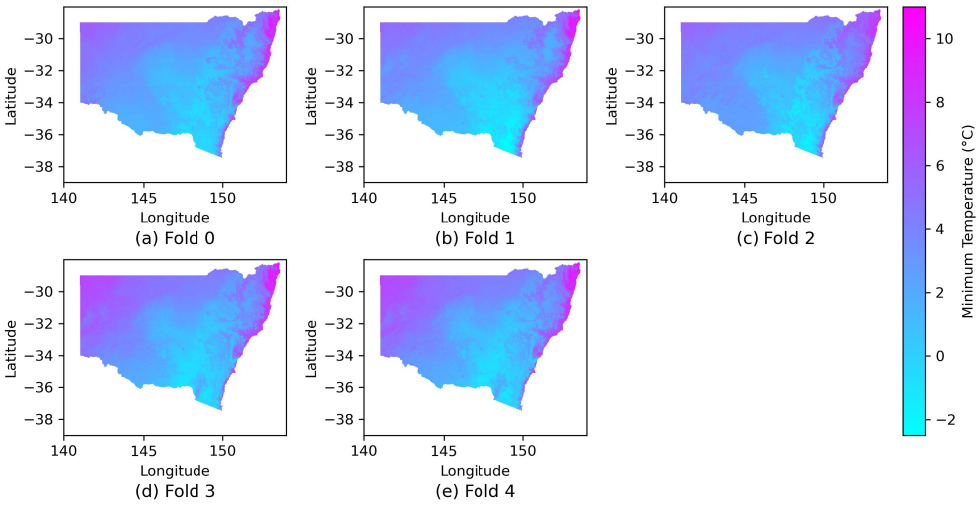


Figure 4.4: Raster Maps from Averaged Results per Fold.

Finally, raster maps (Figure B.4 in Appendix B) created from weighted averages of models within each fold present similar properties to the averaged maps. From the absolute error maps

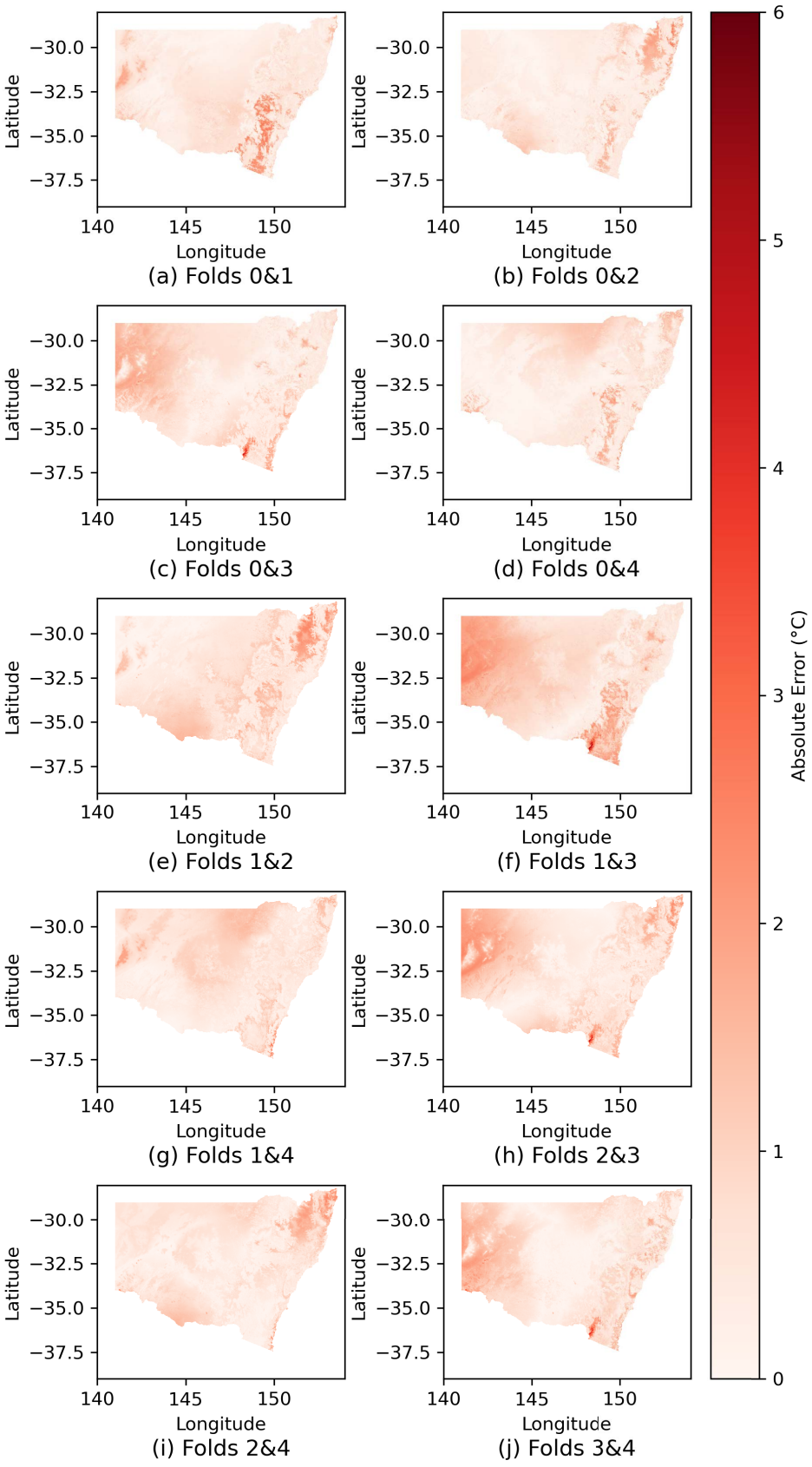


Figure 4.5: Absolute Error Map from Averaged Results Between Folds.

(Figure B.5 in Appendix B), the effect of different training station datasets still persists. The T-tests also present all-zero results.

4.3.2 Model Accuracy

In the second experiment, model accuracy is compared between individual station models for different folds, an averaged station result per fold, a weighted average result of station models per fold, and the baseline (Figure 4.6). The baseline, using on-site sensor data, reaches the highest accuracy with the lowest RMSE. The ensemble methods of averaging and weighted averaging outperform the individual station models in the spatial interpolation-based methods. Ensemble by weighted average reached higher accuracy because higher weights are given to stations that are much similar to the target locations.

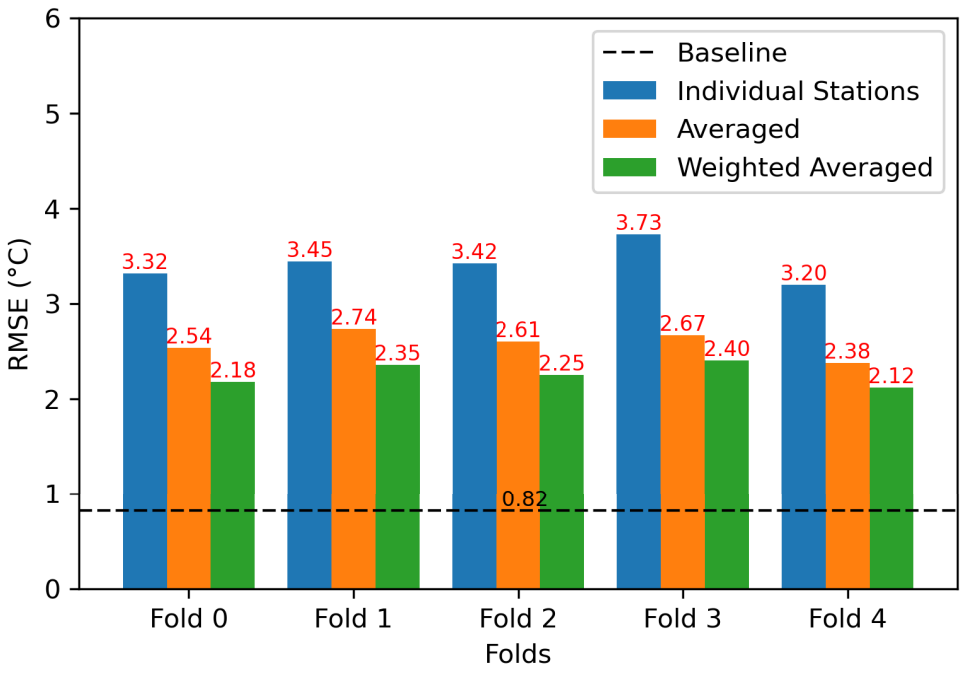


Figure 4.6: RMSE of Individual, Averaged, Weighted Averaged Models Obtained from Year 2017 Testing Datasets.

4.3.3 Effect of the Number of Available Weather Stations

In the final experiment, the ability of fault tolerance of the proposed ensemble methods is tested. The accuracy and event capture rate of the averaged and weighted averaged methods are tested with different numbers of available weather stations. This experiment focuses more on the ability to capture potential events below zero degrees in the future. The mean true positive rate of the

baseline models is 99.62%. To ensure fairness of the experiment, the testing datasets are from the year 2018 (a year after the training datasets).

Figure 4.7 shows the accuracy of the proposed ensemble methods with different numbers of available weather stations 10, 20, 30, 40, 50, and 60. The weighted averaged method outperforms the averaged method with a lower RMSE. Accuracy for both methods increases as the number of stations increases.

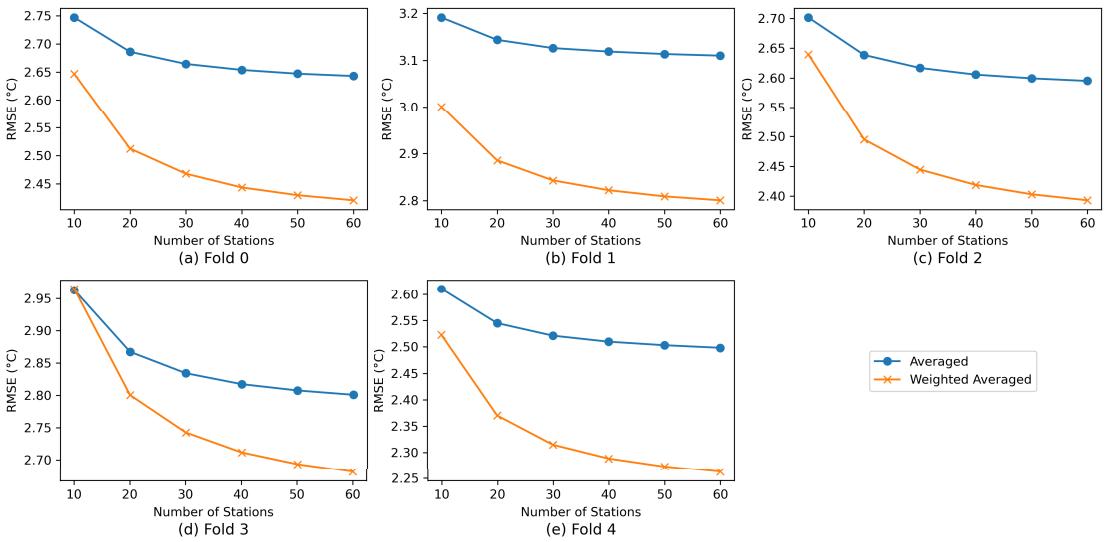


Figure 4.7: RMSE of Averaged and Weighted Averaged Models with 10–60 Available Stations Obtained from Year 2018 Testing Datasets.

Contrary to the accuracy, the event capture rate or true positive rate reduces as the number of stations increases (Figure 4.8). However, the false discovery rate also reduces with the increase of the number of available stations. This illustrates that the event detection rate is high due to high false positives. The ensemble of results with more stations reduces error. However, high errors induce lower temperature predictions, which increase the event capture rate with the number of false positives. Another experiment with the number of available stations 1–10 is conducted to inspect the relationship between true positive rate and false discovery rate.

The RMSE (Figure 4.9) of the ensemble methods for 1–10 stations follows previous patterns. RMSE reduces with the increase of the number of available stations. The true positive rate performs a more extreme pattern. When there is only one available station, the true positive rate could exceed 90% (92.55% as the highest) (Figure 4.10). However, the false discovery rate also exceeds 90% when there is only one available station. In conclusion, as the accuracy of temperature prediction reduces, the models recognize more events. This increase of event recognition increases the true positive rate along with the false discovery rate. The number of stations selected from the ensemble methods should be considering the balance between the event capture rate and number of false positives.

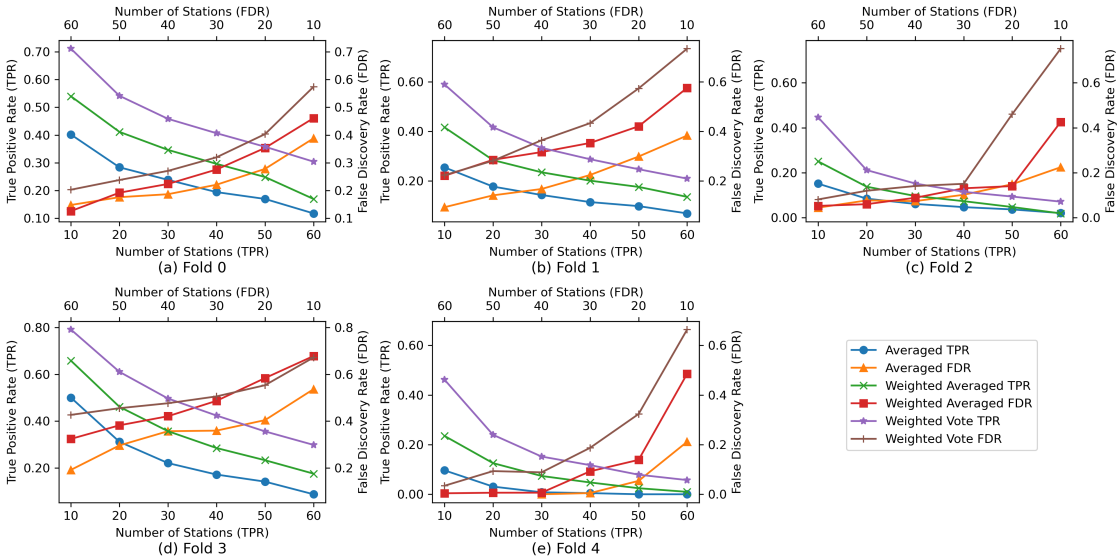


Figure 4.8: True Positive Rate and False Discovery Rate of Averaged, Weighted Averaged, and Weighted Voting Models with 10–60 Available Stations Obtained from Year 2018 Testing Datasets.

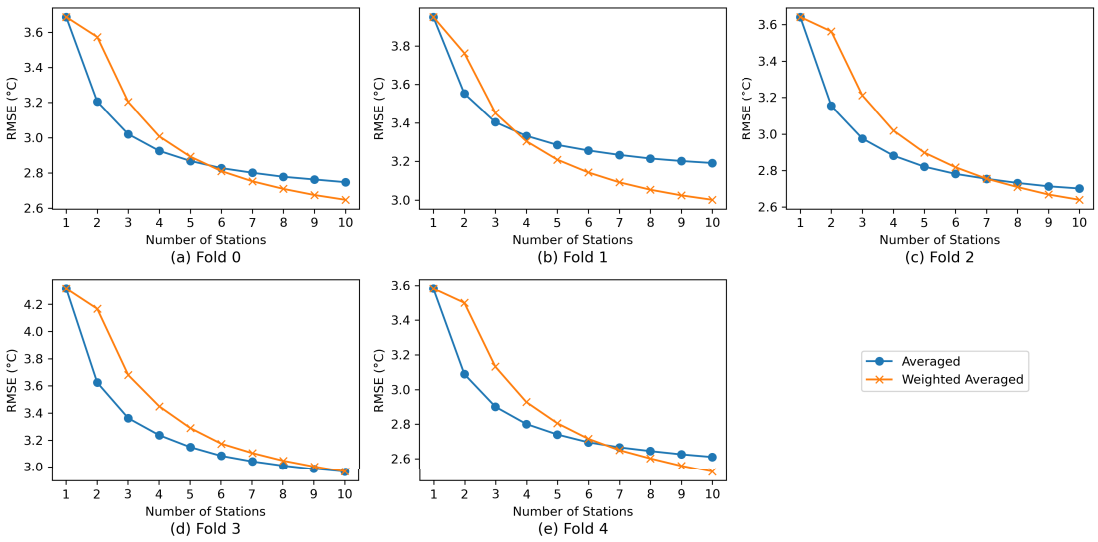


Figure 4.9: RMSE of Averaged and Weighted Averaged Models with 1–10 Available Stations Obtained from Year 2018 Testing Datasets.

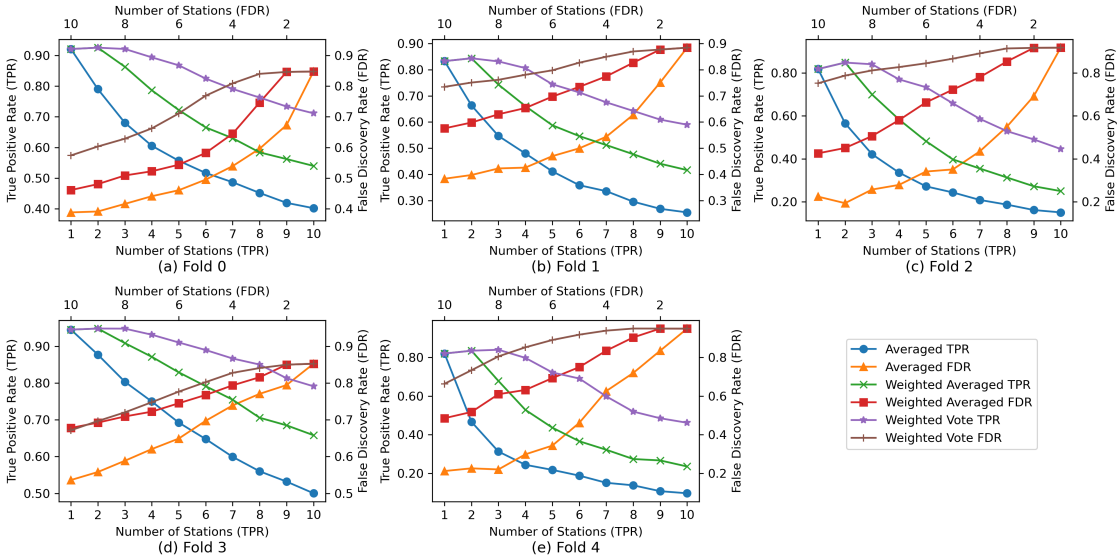


Figure 4.10: True Positive Rate and False Discovery Rate of Averaged, Weighted Averaged, and Weighted Voting Models with 1–10 Available Stations Obtained from Year 2018 Testing Datasets.

Generally, the weighted averaged method demonstrates a higher accuracy and event detection rate compared the averaged method. However, for both methods the true positive rate decreases as the number of stations increases. This phenomenon is due to the low accuracy of the weak predictors and the smoothing effect of the averaging methods. The averaging methods are filtering out the lower extremes of the prediction results. Therefore, a third method or the weighted voting method is proposed. The weighted voting method collects a weighted vote from each of the weak predictors and aggregates the votes. The vote positive when the prediction result is smaller than the triggering temperature zero degrees, else the vote is negative. The weights are computed by the same algorithm of the weighted averaged models.

Figures 4.8 and 4.10 compared the true positive rate and false discovery rate of the weighted voting method with the previous two methods. The true positive rates have significantly increased compared to the previous methods, when utilizing the same number of stations. However, the false discovery rate also increased. To further increase the true positive rate without the increase of false discovery rate of the spatial interpolation-based methods, the accuracy of the weak predictor models should be further improved.

4.3.4 Comparing Proposed Data Aggregation Methods with Traditional Methods

In this subsection, the proposed averaging, weighted averaging, and weighted voting methods are compared with two traditional kriging methods. At each time step, the prediction results of the 60 source weather stations in a data fold are aggregated by IDW and OK to compute the next hour minimum temperatures at the 15 testing weather stations. Similar to the previous subsection, the

number of source stations is variable. The results presented are from Fold 0. From Figure 4.11, the RMSEs of both IDW and OK are lower than the averaging method, but higher than the weighted averaging method. The traditional methods demonstrate similar patterns to the proposed methods in that a higher number of source stations reduces the RMSE.

Similarly, the true positive and false discovery rate trends of IDW and OK are also similar to the proposed methods (Figure 4.12). The true positive rates of the traditional methods are decreasing with the increase of source stations. The false discovery rates are decreasing with the increase of source stations. This shows that the low accuracy of the submodels also impacts the traditional methods. The true positive rates of the IDW method are only higher than the averaging method. In exchange, the false discovery rates of IDW are only higher than the averaging method. The true positive rates of OK are close to the weighted voting method. When there are 10 and 60 source stations, the true positive rates of the weighted voting method outperform OK. In exchange, the false discovery rates of OK at 10 and 60 stations are lower than the weighted voting method. In general, most of the proposed methods demonstrated a better ability to capture frost events than IDW. However, IDW demonstrated a lower false discovery rate. On the other hand, OK outperformed most of the proposed methods with a higher true positive rate except for the weighted voting method. However, OK also has a relatively high false discovery rate. Overall, the traditional methods still have the potential to be used as data aggregation methods for the submodels.

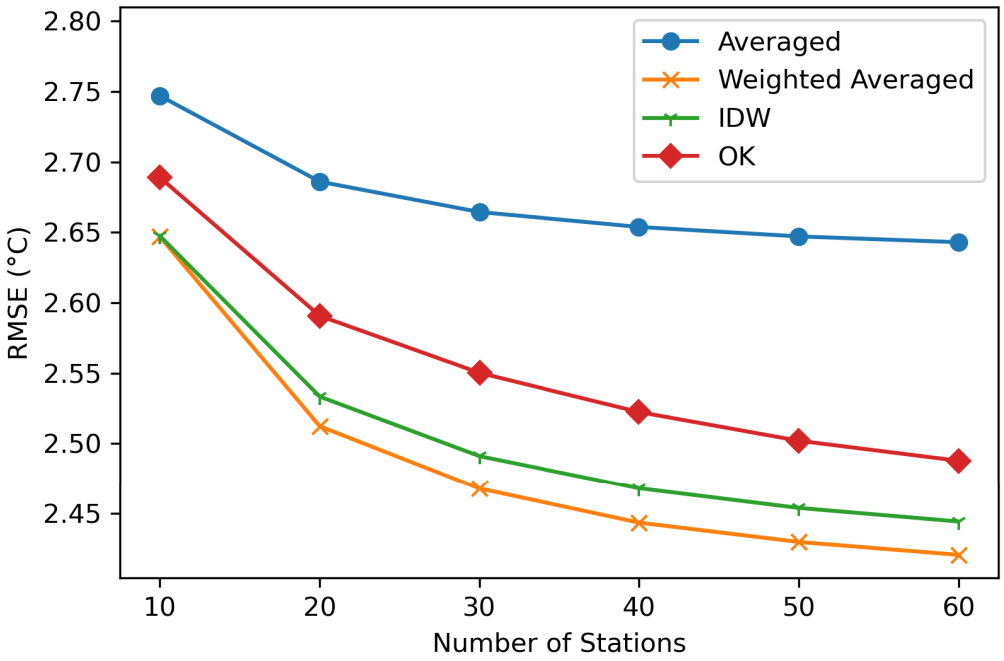


Figure 4.11: RMSE of Averaged, Weighted Averaged, IDW, and OK Models on 2018 Fold 0 Data.

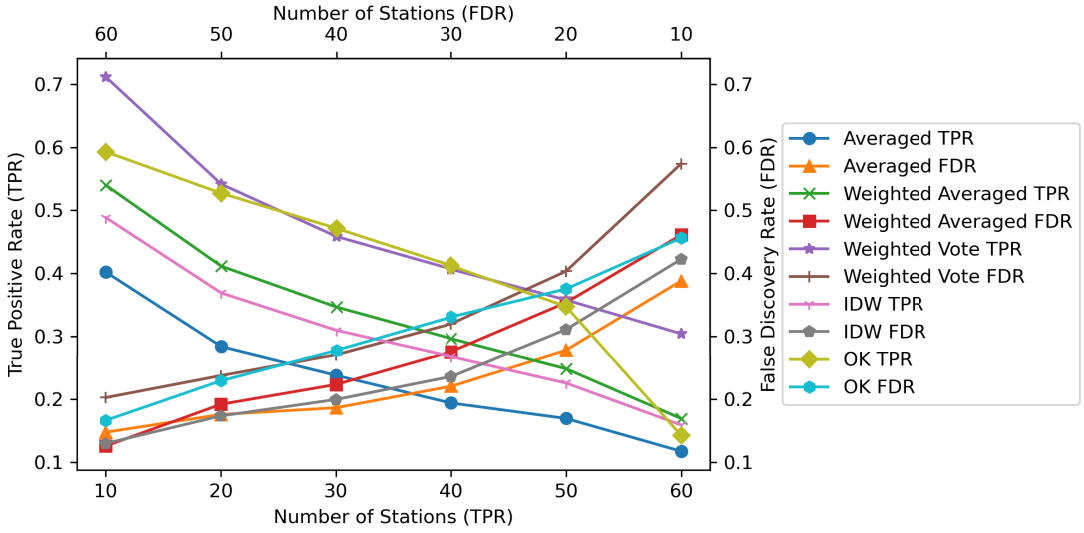


Figure 4.12: True Positive Rate and False Discovery Rate of Averaged, Weighted Averaged, Weighted Voting, IDW, and OK Models on 2018 Fold 0 Data.

4.3.5 Limitations and Open Challenges

After testing the proposed method with the baseline, there are a few limitations of the method detected. These limitations are open challenges that lead toward future development of spatial interpolation-based frost prediction methods.

4.3.5.1 Further improvement of the accuracy

The errors of the proposed method are higher than the baseline created from on-site datasets. The highest true positive rate (or event capture rate) achieve from the proposed methods is 92.55%, which is lower than the average true positive rate (99.62%) of the baseline models. Moreover, this high rate of the proposed method is only achieved with high numbers of false positives. To increase capture rate and decouple event capture rate with the number of false positives, the increase of accuracy is inevitable. With the current accuracy, spatial interpolation-based methods cannot fully replace previous methods based on on-site historical datasets and sensors.

4.3.5.2 Lacking of ground truth to validate models

As spatial interpolation-based models are constructed with five-fold validation, the testing datasets are limited [286]. In this chapter, the testing datasets are generated from 15 weather stations for each fold. Therefore, any accuracy metrics can only represent the accuracy on these 15 weather stations. In the production environment, predictions are required on other locations. The model accuracy on other locations is uncertain. In future works, more weather stations could be involved to reduce this uncertainty.

4.3.5.3 Models are highly sensitive to the choice of training weather stations

As shown by experiment 1, raster maps produced from models of different folds are significantly different from each other. The models are affected by choices of training weather stations. From experiment 2, this choice also affects the accuracy of models. In future model construction with cross-validation, the accuracy of each fold should be carefully examined. A possible method to mitigate this issue is the increase in the number of weather stations. As the number of stations increases, the effect of individual stations reduces.

4.3.5.4 Implementation with other spatial interpolation methods

The focus of this paper is to eliminate the dependency on local sensors and data in frost prediction applications. Therefore, the proposed method is developed from previous machine learning-based frost prediction methods. However, there are also other factors that are not considered. The first factor is model design. Each submodel of the proposed method is an end-to-end model that directly produces the output minimum temperature of a target location. Could this be designed into a multi-stage pipeline model? For example, spatial interpolation is applied to the environmental parameters of a source station to obtain the parameters at the target location. Then, these environmental parameters could be fed to a local sensor/data-based prediction model to obtain the minimum temperature. These parameters could be reused for other applications at the target location. Another unconsidered factor is the usage of other existing spatial interpolation methods. In this paper, the test results of OK demonstrated some potential. There are other kriging methods. These methods are reviewed in [287]. The authors of [287] also stated other machine learning models used in spatial interpolation methods. These models are support vector machine, random forest, and neuro-fuzzy network. Other than these, there are also state-space models based on Bayesian inference [288, 289].

4.3.5.5 Application of similar methods to other domains

The recent frost prediction methods mentioned in Section 4.1 all predicts the local condition with local historical climate data-based models. Data availability is an inevitable factor to consider when constructing these models. Some other domains using local historical data could also be impacted by data availability. Therefore, these domains could also benefit from the proposed method in this chapter. Domains such as forest fire forecasting [290] and soil property prediction [291] that are evolving with spatial methods could be further explored with a varied version of the proposed method. Other fields such as rain forecasting and atmospheric weather forecasting [292] could implement the proposed method as an exploration of a fault-tolerance design.

4.4 Summary

This chapter proposes a spatial interpolation-based frost prediction method. As a precondition to answer **Research Question 5**, this method aims to eliminate the dependency on on-site historical datasets/sensors during model training, validation, testing, and future operations. The proposed methods are ensemble learning methods based on ANN models. The two ensemble methods are averaging and weighted averaging of weak predictors. Each weak predictor is constructed using one weather station as the climate data source. The baseline models are ANN models trained with on-site historical data. There are three experiments conducted to test the performance of models. The first experiment compares the raster map outputs of spatial interpolation-based models. Raster maps are constructed from the individual weak predictors, averaged results of weak predictors, and weighted averaged results of weak predictors. The results of the T-tests show that the raster maps for different folds are significantly different from each other. This shows that the models are significantly affected by the training datasets. Hence, the division of folds is important. The second experiment shows that the weighted averaged method provides the lowest error among the spatial interpolation-based methods. After that, the final experiment reveals the effect of the number of available stations. Accuracy increases as the number of available stations increases. However, the event capture rate increases with the reduction of station numbers. This increase is related to the increase in false positives. Apparently, higher errors induce the triggering of more events, which increases both event capture rate and false positives. The experiments indicate three limitations of the work. The first limitation is accuracy. Accuracy needs to be increased to eliminate the relationship between high event capture rates and high false positives. Also, as a spatial interpolation-based method constructed by five-fold validation, only 15 stations per fold act as the ground truth to test the models. Finally, the models are highly sensitive to the choice of training weather stations. This uncovers the importance of fold selection. In conclusion, as a standalone system, the proposed spatial interpolation-based frost prediction method could capture frost events. However, this system is still limited by high false positives and the lack of stability. Therefore, The system is not sufficient to fully answer **Research Question 5**. It could be deployed as an alternative system when on-site historical datasets are temporarily unavailable. In the next Chapter (Chapter 5), a response to **Research Question 5** is further explored. The spatial interpolation-based system is integrated as a fault tolerance mechanism besides a local sensor-based system.

5

A Frost Forecasting Framework Featuring Spatial Prediction-based Stopping Mechanism

5.1 Introduction

The recent frost prediction and protection systems reviewed in Chapter 2 often rely on IoT technologies with wireless sensors and protection equipment placed on the site [34]. However, as these systems depend solely on a single sensor mote, fault tolerances of these systems are questionable [34]. Also, as the sensors are placed together with the protection equipment, activation of this equipment could interfere with the outputs of local sensors (Chapter 3 [257]). These issues, along with **Research Question 5**, have been addressed in the previous chapter (Chapter 4). However, the system proposed in the previous chapter suffers from instability and low accuracy (the highest detection rate is only 92.55%). Further improvements in the reliability and accuracy of frost prediction and protection models are required. The frost prediction framework described in this chapter complements the response to **Research Question 5** provided by Chapter 4. The use case scenario of the framework is assumed to be after the operation of a spatial interpolation-based system. During the operation of the spatial interpolation-based system, local data could be collected to update the system into the proposed framework.

This chapter proposes a near real-time frost prediction framework to enhance the reliability of frost protection systems. This framework combines a relatively high accuracy prediction method (Chapter 3 [257]) based on local sensors (local prediction module or LPM) with a model (Chapter 4 [293]) based on weather station measurements (remote prediction module or RPM). Both models predict the minimum temperature of a future time period. The RPM is integrated to pre-

vent interference from local protection equipment and provide extra fault tolerance. The stopping mechanism controlling the frost protection equipment should also be based on the results from a remote source uninfluenced by the protection equipment itself.

5.1.1 Related Work

IoT-based frost protection systems started as an assistant to human monitoring [31]. With this method, temperature sensors are used to wake up the farmer during night. The farmer monitors the temperature manually to decide the triggering of protection equipment by experience and leave the equipment on until the next morning. After that, modern implementations [226, 229–231] of the system are automated with IoT systems and WSNs (Table 5.1). These frost protection systems rely on one or two local temperature sensors. An early implementation of these systems is equipped with a smoke generator/burner as the protection mechanism against frost on an open field [229]. Temperature-based fuzzy logic rules are applied to control the output magnitude of the protection mechanism. However, the stopping condition of the protection equipment is not discussed. This design is improved in [231]. The energy requirement of the burner is minimized by limiting the triggering threshold. A similar stopping mechanism is developed in [230]. The sprinklers in the fields are stopped when the temperature reaches a threshold. Another type of stopping mechanism is created with a timer [226] and the timer stops the mechanism in the morning. Some system fault tolerance exists with the dual sensor design. At this stage, most systems determine the stopping condition with near-real-time mechanisms. However, the reliability of these mechanisms is questionable as the sensor nodes are placed near the protection equipment. Contamination of sensor node readings is never handled. Additionally, as the systems rely solely on one or two sensors, there is limited system fault tolerance. Therefore, external data sources may be required.

More recent frost protection system architectures [16, 232] included more sensor types. It is significant that these architectures recognize external data sources. These external data sources are often drawing data from nearby weather stations (Table 5.1). The authors of [16] tried to predict temperature with the aid of a local sensor and weather forecast service. This method shows a high error of over 9 degrees. As this work only provides prediction, the stopping mechanism is not implemented. A more detailed system is proposed in [232]. The relationship between the site of interest and nearby weather stations is studied. A frost index is designed to predict the occurrence of frost. However, this method only provides daily predictions. The protection mechanism would need to be operating for the whole night without shorter prediction windows and stopping conditions.

In summary, earlier systems need to be extended to handle the effect of local protection mechanisms on the local data source. Additionally, system fault tolerance is limited, with the reliance on only one or two sensors. On the other hand, operational resources are preserved with the inclusion of near-real-time stopping mechanisms. However, this mechanism has been removed in more recent systems. Recent systems predict frost with the aid of external weather stations. This provides some fault tolerance to the systems. However, as the local sensor readings are still a

Table 5.1: Recent Works on Frost Protection Systems.

Year	**Local Data Source**	**External Data Source**	**Stopping Condition**	**Fault Tolerance**	**Detection Frequency**
2008 [229]	Temperature sensors	N/A	Equipment output is only adjusted by fuzzy logic	N/A	Hourly
2009 [230]	Thermistors	N/A	Equipment stopped by reached temperature	N/A	Second-wise
2009 [226]	Air temperature sensors	N/A	Stopped in the morning by a timer	Some tolerance provided by dual sensor design	Per 10 minutes
2012 [231]	WSN temperature sensors	N/A	Equipment stopped by reached temperature	N/A	Per 15 minutes
2017 [16]	WSN temperature sensors	Public weather forecast service	N/A	Calibration with weather forecast	Per 15 minutes
2019 [232]	WSN sensors	Weather station service	N/A	Calibration with nearby weather station	Daily
Proposed Framework	**WSN sensors**	**Weather station data, Satellite data**	**External stopping condition independent of local sensors**	**Calibration from independent multiple weather stations**	**Minute-wise**

dominant part of the prediction input, this fault tolerance is limited.

The proposed framework aims to include both the near-real-time stopping condition and fault-tolerant external data source designs from earlier methods and enhance these designs with each other. The proposed framework includes both local and external data sources. The stopping mechanism is triggered mainly by the prediction readings from external data sources to eliminate the contamination from local protection equipment (e.g., heaters and sprinklers). As the external data sources are not likely to be affected by the temperature or humidity change from the local protection equipment, the system can still provide uncorrupted outputs. Based on these outputs, a stopping mechanism for the protection equipment can be operated. With this real-time stopping mechanism, the operation time of the protection equipment can be minimized. Since prediction systems only with external data sources are less accurate than using local sensors [16, 293], the external data sources are only used to provide a stopping condition and extra fault tolerance when the local sensors readings are contaminated or unavailable. To test the proposed framework, a version of the framework is implemented with the system developed in Chapter 3 [257] as the LPM and the system developed in Chapter 4 [293] as the RPM. The contributions of this chapter are:

1. Proposing a frost prediction system framework.
2. An implementation of the framework with existing local and remote prediction methods.
3. Eliminating the unavailability of frost prediction systems caused by local sensor faults.
4. Establishing a model-based stopping mechanism for frost protection equipment.

The rest of the chapter is organized into three sections. Section 5.2 describes the methodology, which includes the framework structure, data sources, and experiment settings. Section 5.3 presents the results and discusses the insights from the results. Finally, the chapter is summarized in Section 5.4.

5.2 Methodology

This section describes the methodology involved in constructing the proposed prediction method. First, the model structures are illustrated with the designed data inputs. Then, the data sources and data types of these data inputs are revealed. Finally, the experiment methods are described.

5.2.1 Proposed Framework Structure

In addition to the existing prediction methods based on local sensors, the proposed method included a prediction model based on external weather stations to act as a fault-tolerance mechanism and a stopping mechanism for the frost protection system. Therefore, the proposed framework structure consists of three modules. These modules are the original LPM, the additional RPM and the merging module (MM) to merge the results from the prediction modules (Figure 5.1).

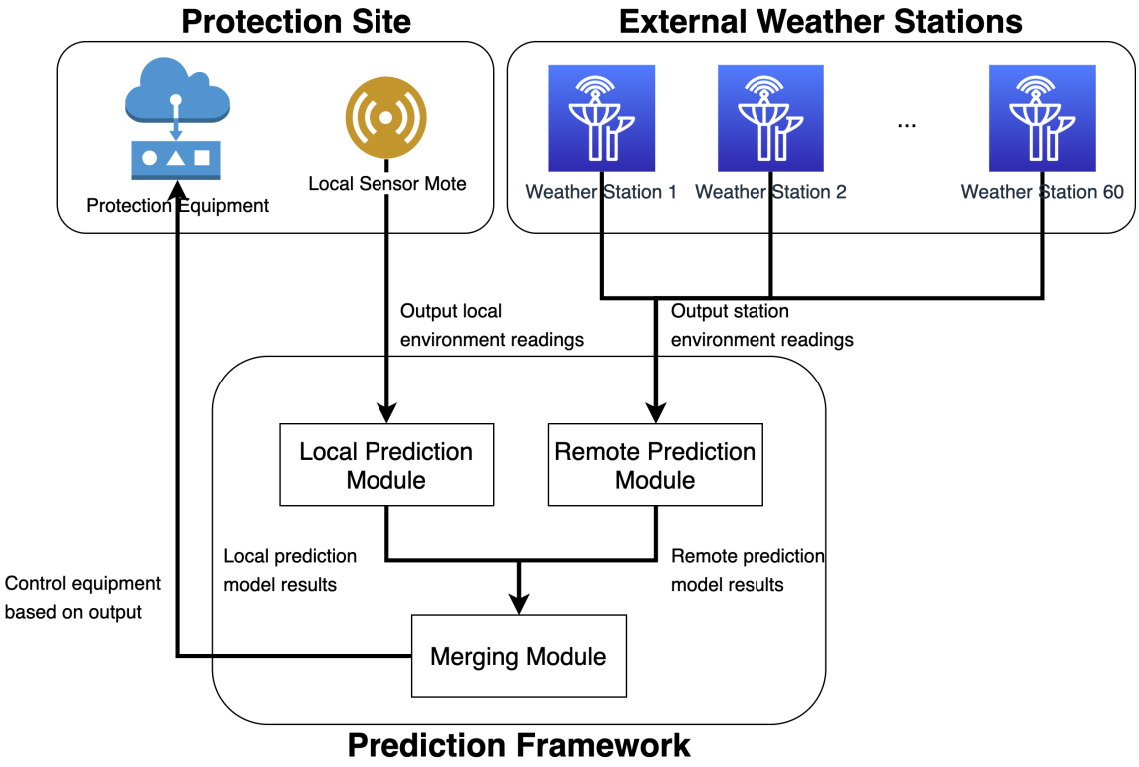


Figure 5.1: Experimental Implementation of the Proposed Prediction Model.

The LPM is a model that provides minute-wise predictions of the next hour minimum temperature. As suggested by the module name, these predictions are obtained from local sensor inputs. In this chapter, a neural network model similar to the model in Chapter 3 [257] is implemented as the LPM. The model accepts current temperature and relative humidity data and outputs the next hour minimum temperature.

Since the status and availability of the local sensors could be affected by multiple factors such as mote energy consumption, hardware fault/failure, and activation of heating equipment [293], the RPM predicts the next hour minimum temperature from data obtained from external sources to ensure the availability of the prediction system, while the local sensors are unavailable. In the experiments, the spatial prediction method from Chapter 4 [293] is used as the RPM. This model involves a series of sub-models. Each of the sub-models uses data from a fixed weather station to predict the next hour minimum temperature of a remote location. Each of the sub-models accepts climate data (temperature, dew point, relative humidity, wind speed and direction) and location features (longitude, latitude, digital elevation and normalized difference vegetation index). For sub-model inferences, location features of both the weather station and the target location are required. Climate data is only needed from the fixed weather station. With multiple sub-models, the fault-tolerance of the entire remote prediction model is high. In the implementation, every remote prediction is supported by at most 60 weather stations.

The MM processes the outputs from the prior two modules to ensure the availability of the

system. By the control and management of the MM, frost protection equipment can be stopped at a suitable time to minimize operational costs. In this chapter, the Kalman filter and the particle filter are tested as merging algorithms. Both filters receive outputs from the LPM. The filters will switch to receive RPM outputs under the below two conditions:

1. A current, real value is unavailable from the LPM. **OR**
2. The frost protection equipment is switched on.

The first condition indicates possible malfunctioning of the local sensor. When the second condition is true, the frost protection equipment could corrupt the local sensor values. In both conditions, the local sensor cannot be trusted. Therefore, the filters will switch to accept outputs from the RPM.

The Kalman filter is set to accept simple one-dimensional temperature measurements. To construct the Kalman filter, the process noise covariance, observation noise covariance, state-transition model, and observation model are calculated. The process noise covariance and observation noise covariance are obtained as variances because these are calculated from one-dimensional data. The process noise data for each target location is obtained by subtracting the average minimum temperature from the minimum temperature. This assumes a stable minimum temperature over time. All changes in the minimum temperature are viewed as noises by the Kalman filter estimation. Therefore, the state-transition model is set to one. There are two sets of observation noise variance corresponding to the LPM and RPM. During operation, when the filter switch to accept outputs from the LPM or RPM, the observation noise variance also switches. The difference between predicted minimum temperatures from an LPM or RPM, and the true minimum temperatures computes the observation noises for each target location. Finally, since the observation and true state are in the same space, the observation model is also set to one.

The particle filter requires the process noise covariance, observation noise covariance, and the number of particles. The covariances or variances are the same as the Kalman filter. The number of particles is set according to the experiment condition defined in Section 5.2.3.

5.2.2 Data Sources

The study area of the experiments includes the New South Wales and Australian Capital Territory states of Australia. There are two sets of data sources obtained from the study area. The locations (Appendix A) of the data sources correspond to 75 different weather stations. The first set of data sources is used in the training and testing processes of the local and remote prediction modules. Table 5.2 outlines the data sources.

The second set of data sources includes the sunrise and sunset times of the year 2018. This data is generated from the geodetic calculators of Geoscience Australia [294]. The geographical coordinates of the 75 weather stations are inserted into the calculators. The sunrise and sunset times of June, July, and August are later used to form a baseline in the experiments.

Table 5.2: Climate and Location Data Sources from Different Organizations

Organization	**Data Types**
Bureau of Meteorology [261]	Temperature, dew point, relative humidity, wind speed and direction
Geoscience Australia [281]	Digital elevation
NASA [282]	Normalized difference vegetation index
Department of Industry, Science, Energy and Resources [283, 284]	New South Wales and Australian Capital Territory state boundaries

5.2.3 Experiments

The models constructed for the experiments are trained with climate data and location data (first three data sources of Table 5.2) of the 75 weather stations obtained for the year 2017. Then, the testing results are obtained with data representing the year 2018. There are two sets of experiments performed to test the proposed prediction method. The first set of experiments compares the model accuracy of merging methods. For the proposed methods utilizing the Kalman and particle filters, to minimize model losses, it is assumed that there is no sensor value corruption from the protection equipment and the remote prediction outputs are only activated when the local prediction outputs are not available. Also, the local sensor motes are able to transmit a set of readings each minute. The particle number of the particle filter is set as 10000 to balance model accuracy and process power requirements.

The second set of experiments compares model accuracy at different transmission periods of the LPM. The transmission period of the local sensor motes is variable. The transmission period is set to one minute, five minutes, ten minutes, fifteen minutes, twenty minutes, half an hour, and an hour. It is also assumed that the protection mechanism will corrupt local sensor readings. The MM uses spatial prediction module outputs when local sensor data is unavailable, between local reading transmissions, and during operation period of protection mechanisms.

Other than the model losses, the second set of experiments also compares the ability of frost event detection. This involves the construction of the baseline. The baseline utilizes the ideal state of the traditional frost detection method. Recent prediction methods are predicting frost for the next 12–24 hours and the more common frost type (radiation frost) appear during night times [200]. Therefore, in our modeled baseline, when there is one reading after the sunset below a certain temperature threshold, all times after sunset and before the next day sunrise are considered as frost events. With the baseline, the true positive rates, false discovery rates, and the frost protection equipment operation times of the proposed methods are analyzed. The conditions of the experiment sets are summarized in Table 5.3.

Table 5.3: Experiment Conditions

Experiment Set Number	**Particle Number (for particle filter)**	**Transmission Period (Minutes)**
1	10000	1
2	1, 10, 100, 1000, 10000	1, 5, 10, 15, 20, 30, 60

5.3 Results and Discussions

This section outlines the results obtained from the two sets of experiments. Model accuracy is first compared between the Kalman filter and the particle filter. Then, the ability of frost detection is evaluated for models with different local sensor transmission rate. After that, operation time of the frost protection equipment is analyzed. Finally, the results lead to discussions of limitations and future works.

5.3.1 Accuracy of Different Merging Methods

Table 5.4 is collected from the first set of experiments mentioned in Section 5.2. The mean RMSEs from the Kalman filter and particle filter (10000 particles) are quite close to the sole LPM. The particle filter has a insignificantly higher loss than the sole LPM and Kalman filter.

Table 5.4: Mean RMSE of the Sole LPM and Different Result Merging Methods with One-minute Transmission Period and without Assumed Influence from Frost Protection Equipment.

Method	**Mean RMSE** ($^{\circ}C$)
Sole LPM	0.9733
Kalman Filter	0.9733
Particle Filter	0.9742

Figure 5.2 is collected from the second set of experiments. The transmission period is also set to one minute to minimize the loss. Differ to the first set of experiments, the influence of the frost protection equipment to the local sensors is considered. Therefore, outputs from the RPM is processed by the MM when the frost protection equipment is in operation. From Figure 5.2, the mean RMSE reduces significantly for particle numbers 1 and 100. However, with the particle number 1000, this reduction is insignificant (less than 5%). The mean RMSE of the particle filter with 1000 particles is $1.3855^{\circ}C$, which is still higher than the mean RMSE of the Kalman filter ($1.3831^{\circ}C$). When the particle number reaches 10000, the mean RMSE is $1.3690^{\circ}C$, which is slightly better than the Kalman filter. However, considering the significant amount of processing power required for particle filters with more than 100 particles [295] and the insignificant amount of difference between Kalman filter results. The Kalman filter is in favor. However, to ensure the completeness of the experiments, the results of the particle filter with 10000 particles are also presented in the subsections below.

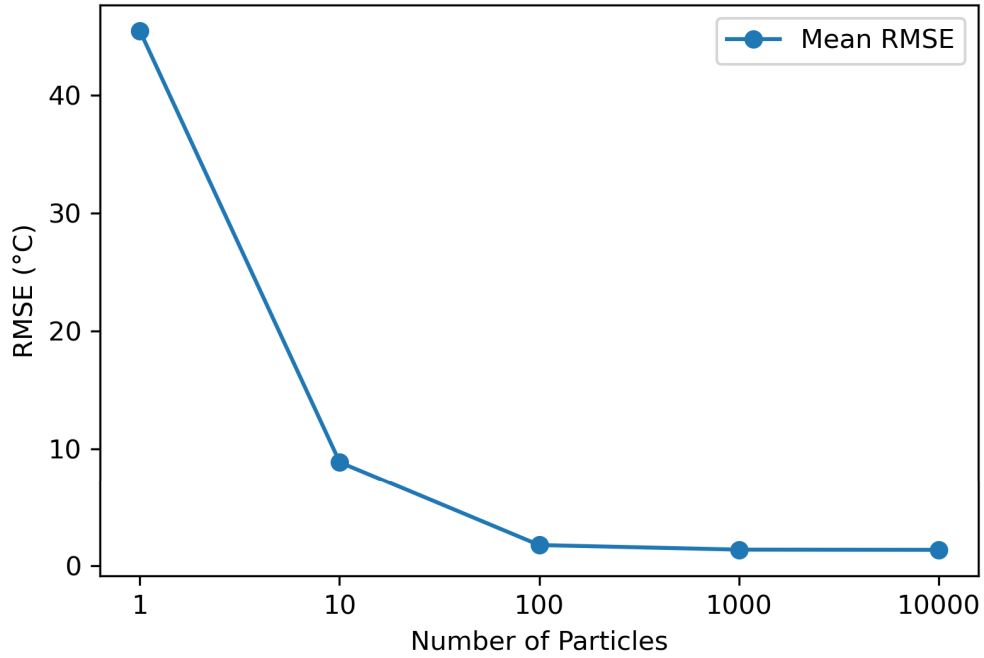


Figure 5.2: Mean RMSE of the Particle Filter Merging Method with One-minute Transmission Period and Variable Number of Particles.

5.3.2 Frost Detection with Different Transmission Periods

In this subsection, the proposed frost prediction method is compared with a baseline. The baseline is the assumed traditional frost detection method, in which the protection equipment is switched on after sunset if any event is predicted. A temperature measurement below $0°C$ at a certain minute is defined as an event in the experiments. The protection mechanism will be operating if such an event is predicted. Table 5.5 describes the baseline results to compare with the proposed methods. Since the baseline does not produce numeral values for prediction, it is unnecessary to describe it with the RMSE. The true positive rate, false discovery rate, and operation time are sufficient to be compared with the proposed method.

Table 5.5: Baseline Results.

Mean True Positive Rate	**Mean False Discovery Rate**	**Mean Operation Time (Hours)**
0.8127	0.6982	208.24

To compare the effect of the local sensor transmission period, the mean RMSE is presented in Table 5.6. For both merging methods, the mean RMSE increases as the transmission period increases. This is caused by the increasing usage of the RPM results with the increase of transmission period. With more usage of the remote prediction results, the mean RMSE also approaches

the relatively higher values of the spatial prediction models [293]. There is a significant increase of mean RMSE between transmission period of one and five minutes. After that, the mean RMSE increases gradually. This is because at transmission period of five minutes most prediction results adopted by the merging method are from the remote source (greater than 80%). The model will be more accurate with a lower transmission period. From an inter-method perspective, the mean losses do not demonstrate significant differences between the Kalman filter and particle filter with 10000 particles.

Table 5.6: Mean RMSE ($^{\circ}C$) of Different Result Merging Methods with Variable Transmission Periods and Assumed Influence from Frost Protection Equipment.

Period (Minutes) / **Method**	1	5	10	15	20	30	60
Kalman Filter	1.3831	2.3330	2.4229	2.4519	2.4667	2.4812	2.4960
Particle Filter (10000 Particles)	1.3690	2.3309	2.4226	2.4521	2.4671	2.4818	2.4968

The mean true positive rate of the proposed method is compared with the baseline in Table 5.7. Similar to the mean RMSE, there is no significant difference between the Kalman filter and particle filter with 10000 particles. The mean true positive rate decreases as the transmission period increases. When analyzed with the mean RMSE (Table 5.6), the mean true positive rate decreases as the loss of the models increases. Clearly, with a higher model accuracy, the detection rate is higher. Finally, the bold values on the table reveals that both merging methods, with local sensor transmission period of one, five, ten, fifteen minutes, capture more true frost events than the baseline.

Table 5.7: Mean True Positive Rate of Different Result Merging Methods with Variable Transmission Periods and Assumed Influence from Frost Protection Equipment (Bold values are better than the baseline).

Period (Minutes) / **Method**	1	5	10	15	20	30	60
Kalman Filter	**0.9987**	**0.9460**	**0.8883**	**0.8386**	0.7950	0.7197	0.5735
Particle Filter (10000 Particles)	**0.9987**	**0.9466**	**0.8900**	**0.8408**	0.7985	0.7242	0.5805

The mean false discovery rate shows the amount of false positives in all positives recognized by the models. This rate decreases as the transmission period increases (Table 5.8). All mean false discovery rates of the proposed models are lower or better than the baseline.

Per site analysis of the results is conducted to compare the true positive rates and false discovery rates of the merging methods with the baseline site by site. Figure 5.3 demonstrates the comparison of true positive rates per site between the Kalman filter and baseline. The circles represent a site that the Kalman filter results have a higher (better) true positive rate than the baseline.

Table 5.8: Mean False Discovery Rate of Different Result Merging Methods with Variable Transmission Periods and Assumed Influence from Frost Protection Equipment (All values are better than the baseline).

Period (Minutes) / **Method**	1	5	10	15	20	30	60
Kalman Filter	0.6300	0.5488	0.5069	0.5049	0.4792	0.4640	0.4370
Particle Filter (10000 Particles)	0.6377	0.5567	0.5127	0.5110	0.4839	0.4672	0.4390

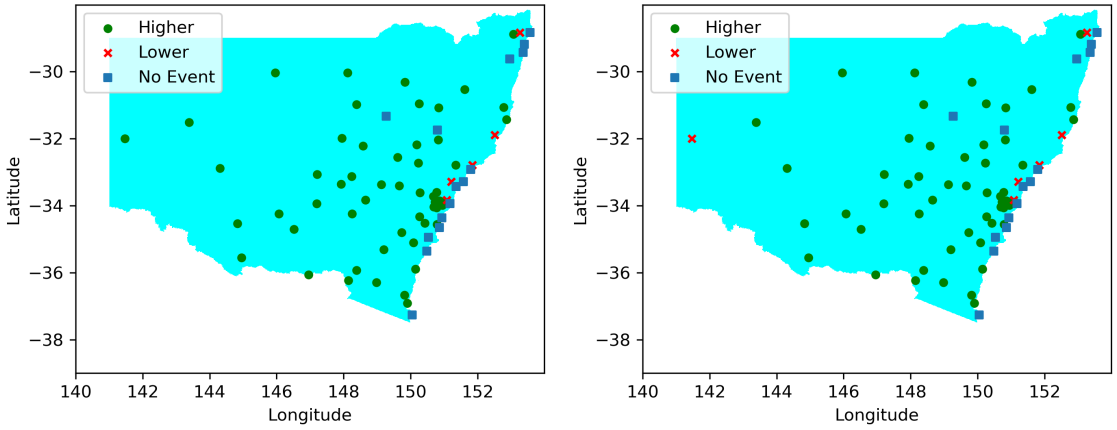
Table 5.9: Mean Operation Hours of Different Result Merging Methods with Variable Transmission Periods and Assumed Influence from Frost Protection Equipment (Bold values are better than the baseline).

Period (Minutes) / **Method**	1	5	10	15	20	30	60
Kalman Filter	208.74	**162.16**	**139.33**	**131.01**	**118.08**	**103.85**	**78.79**
Particle Filter (10000 Particles)	213.19	**165.15**	**141.27**	**132.98**	**119.66**	**105.13**	**80.02**

The crosses symbolize a site with a lower true positive rate than the baseline. Finally, squares demonstrate sites with no events. In Figures 5.3a - 5.3c, most sites shows a higher true positive rate than the baseline. However, there is a significant increase of sites with a lower true positive rate than the baseline when the transmission period approaches 15 minutes (Figure 5.3d). For transmission periods of 20, 30, and 60 minutes, there are also more sites with a lower true positive rate than the baseline (Not shown in figures as the results are similar). This result is partially consistent with Table 5.7. However, it uncovers that even the mean true positive rate is higher than the baseline for the merging methods with a transmission period of 15 minutes, it is still not suitable as more sites are showing a lower true positive rate than the baseline. The analysis results of the false positive rates are also not included as they do not present new insights.

5.3.3 Operation Time of Protection Equipment

The final set of measurements obtained from the second set of experiments draws in the operation hours of the protection equipment. Table 5.9 shows that the operation hour is inversely proportional to the transmission period. This result is consistent to Tables 5.7 and 5.8. The reduction of operation time is caused by the reductions of true positive and false positive events. Table 5.9 also shows that both merging methods with transmission periods of 5, 10, 15, 20, 30, and 60 minutes are shorter than the baseline. With a transmission period of one minute, the transmission hours are insignificantly longer (less than 5%) than the baseline. However, considering that more true frost events are captured than the baseline, it is still acceptable for the one-minute transmission time method to have an insignificantly longer protection equipment operation time (Figure 5.4). Figure



(a) 1 Minute

(b) 5 Minutes

(c) 10 Minutes

(d) 15 Minutes

Figure 5.3: Per Site Comparison between the True Positive Rates of the Kalman Filter results and the Baseline with Transmission Periods of 1, 5, 10, 15 minutes.

5.4 analyzes the composition of the operation time of the baseline and the Kalman filter (Particle filters are not included as they draw similar insights). This figure also split the operation times as responses for true positive and false positive events. On average, the baseline uses 69.82% of operation time on false positives. The Kalman filter-based methods invest less proportion of time on false positives than the baseline. However, the rate of operation time spent on false positives are over 50% for models with transmission periods of 1, 5, 10, and 15 minutes. The lowest proportion of time on false positives is 43.70%. In conclusion, the operation time compositions of the proposed methods are superior to the baseline for their similar or lower operation times and lower proportion of time over false positives. However, the operation time spent on false positives are still high and should be reduced in future works.

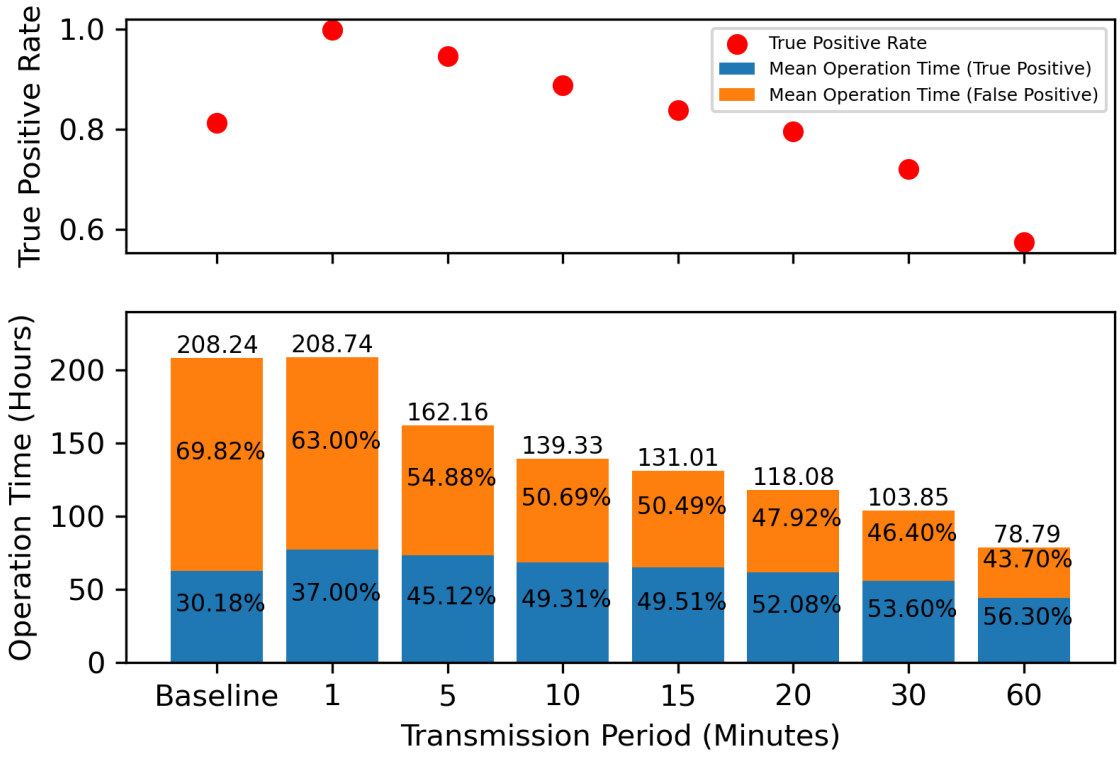


Figure 5.4: Mean Operation Hours and True Positive Rate of the Kalman Filter Merging Method with Variable Transmission Periods.

5.3.4 Limitations and Open Challenges

The experiment results show that the Kalman filter is preferred for the implementation of the proposed framework due to similar losses and fewer processing requirements compared to the particle filter. Both methods have higher, but close losses to the sole LPM models. The comparison between the proposed models, using variable transmission periods, and the frost detection baselines reveals that transmission periods of 1, 5, and 10 minutes are in favor. The proposed model can operate with higher true positive rates and lower false discovery rates than the baseline traditional

method with these transmission periods. After obtaining all the insights, some of the limitations are also revealed by the experiments.

5.3.4.1 Limitations of the Remote Prediction Method

As the RPM dragged down the overall model accuracy in the experiments, it is important to recognize the limitations of the model running within this module. From Chapter 4 [293], limitations of the prediction model based on spatial interpolation are low accuracy, lacking of ground truth to validate models, and high sensitivity to the choice of model training data. These limitations affect the true positive rate, false discovery rate and the stability of the proposed method. As a future direction, models using historical local data to correlate remote prediction methods could be a solution.

5.3.4.2 High number of False Positives

The results show that the proposed method has lower false discovery rates than the baseline. However, most of these rates are over 50%. High false discovery rates lead to unnecessary operation of the frost protection equipment. Compared with the results from Chapter 4, the LPM is a stable source of false discovery rates and should be the priority for improvement. There are two possible methods to lower the number of false positives. The first method is to increase the accuracy of the prediction models. With higher accuracy, the models can detect events with higher precision. The second method is to reduce the prediction window. Each prediction of the proposed method accounts for the next hour. However, the start and the end of a continuous frost event could only take part of the whole hour. Reducing the prediction window to half an hour or less could prevent these false positives.

5.3.4.3 Real-time validation of measurements

This chapter discussed the scenario where the prediction method could fail due to unavailable local sensor data. However, the validity of the data is not considered. All the data involved in this chapter are cleaned and preprocessed. However, in the deployment environment, uncleaned data will be directly fed into the models. This could be a potential source of error that may lead to the failure of the whole system. A possible solution is to develop and include a real-time validation model. Based on historical data, real-time entries could be validated through statistical outlier detection methods and regression data imputation methods [296].

5.3.4.4 Energy Consumption of the Local Sensor Mote

The experiments assumed that the local sensor mote is transmitting data wirelessly from the site of frost protection to a particular processing unit. In IoT applications, radio transmission consumes the highest amount of energy over other usages (processing, sensing, system) [274]. From

the experiment, methods with higher transmission frequencies are providing better results. This indicates that the proposed method requires high transmission frequencies. Higher transmission frequencies will lead to lower battery lifetimes and excess human resources for recharging the energy sources in the deployment environment. A viable solution could be developing a frost prediction method based on edge computing to prevent this high energy consumption from radio transmissions.

5.4 Summary

This chapter proposes a frost prediction framework introducing an extra RPM. The RPM can prevent the effect of protection mechanisms on the prediction model. Also, since the source of data increases, the reliability and fault tolerance increase. There could be multiple remote data sources. Moreover, this RPM can be utilized as the stopping mechanism for frost protection equipment. Including the RPM, the proposed framework contains three modules. The other two modules are the LPM and MM. The LPM provides minute-wise predictions of the next hour minimum temperature using local sensor readings. The MM merges the outcomes from the two prediction modules. The implementations of the framework are then tested in two sets of experiment settings. The experiments are designed to measure the accuracy of the framework with different MM algorithms (Kalmen filter and particle filter) and transmission periods of local sensors. Prediction errors are measured in RMSE. The true positive rate, false discovery rate, and operation time of equipment are also measured and compared against the traditional baseline method. For the merging methods, the results suggest that the Kalman filter is better than the particle filter as the merging method due to the lower requirement of processing power. Compared with the baseline, the models' true positive rates, false discovery rates, and operation times with 1, 5, and 10-minute transmission periods are better than the baseline. This also means more true events discovered, fewer false positives, and fewer resources used on protection than the baseline. However, the accuracy and performance of the proposed methods can be further enhanced by lowering the high false discovery rates. This should be improved in the future with remote prediction models with higher accuracy and further correlation with local readings. Other future improvements to the framework include real-time sensor readings validation, and possible edge computing solutions to reduce energy consumption. Overall, with higher true positive rates than the sole operation of an RPM, this chapter complements the response to **Research Question 5** provided by Chapter 4. In the next chapter (Chapter 6), the limitations of high false discovery rates and energy consumption (**Research Question 6**) of the LPM are discussed.

6

Edge-based Spatially Generalized Frost Prediction with Neural Networks

6.1 Introduction

The experiment results in the previous chapter (Chapter 5) have demonstrated high false discovery rates from frost prediction systems. The LPM produces stable, high numbers of false positives and creates long unnecessary operation time. Therefore, the first priority of this chapter is to improve the LPM. The second priority of this chapter is to indicate the critical energy cost factors of frost prediction systems (**Research Question 6**). Recent frost prediction and protection systems have incorporated IoT technologies, utilizing WSNs [34]. Sensor readings can be wirelessly transmitted to a controller for decision-making. The lifetime or availability of these WSN systems is limited by the energy consumption of the sensor nodes [297]. Therefore, this chapter studies the sensor node energy requirements of a transmission-based system and an edge computing-based system to maximize the lifetime of frost prediction and protection systems.

As indicated in Chapter 2, among the different wireless transmission protocols, LoRa is a suitable technology for frost prediction and protection applications due to its low power consumption and long transmission range [34]. LoRa takes advantage of a modulation technique based on chirp spread spectrum technologies [298]. This technique provides some noise tolerance. Also, LoRa transmissions appear similar to noise, thus making it less susceptible to certain types of security attacks [298]. Overall, the authors of [299] concluded that LoRa provides better connectivity, efficient energy management, security, and less complexity to the IoT-based frost protection systems.

Edge computing is a paradigm placing the processing ability near or co-located with the end devices [300]. This paradigm reduces the transmission latency from the end devices. Other advantages of edge computing are avoided costs on cloud computing, reduced bandwidth requirements and network overhead [301]. In the context of frost protection systems, the application of edge computing offloads the decision-making process to the environmental sensor node. The sensor node assists the triggering of frost protection equipment. Frost protection systems could be independent from a nearby gateway or data center.

In this chapter, a frost prediction method based on edge computing is proposed. Edge computing has the advantage of avoiding the costs on the gateway, data center, and cloud computing [301]. To extend this advantage on low costs, generalized models for frost prediction are introduced in this chapter. As recent frost prediction models are trained with local historical data, systems deployed on new sites need extra deployment time to collect local data for model construction and testing [34,257]. The purpose of the generalized models is to reduce the deployment time by training with existing historical data from weather stations scattered across a region. Also, to evaluate the energy consumption, a LoRa-based application setting is implemented to compare with the edge computing setting.

6.1.1 Related Work

Table 6.1 outlines recent frost protection systems. In early systems, Wireless sensor reading transmission [229, 231] co-existed with wired settings [226, 230]. According to the experimental settings provided, the wireless systems are only designed for small farms. Large farms require a long-range transmission protocol to reach the border gateway. Decision-making for these early systems is based on simple conditions and fuzzy logic with sensor inputs [226, 229–231]. As the performance of frost prediction is either not tested or limited, simple conditions and fuzzy logic are not suitable for frost prediction.

More recent systems predict frost with more complex methods. In [16], a frost index IG is developed with regional weather station data. However, the method requires a centralized server to compute such an index. This places an extra cost on the server.

LoRa is used in [232] to collect data with local sensors. This work also applied regression to predict frost events. As a calibration to local sensors, readings from weather stations are introduced. As this system requests weather station services, a server is required to access the services through RESTful API. The system described in [233] is very similar to the system in [232]. It is also a regression-based frost prediction system, processing readings from local wireless sensors and weather stations through a server.

Compared with the existing systems, the proposed frost prediction system adopts neural networks as models for frost prediction. The neural network is the most accurate model type among recent frost prediction methods [34]. The proposed system also utilizes edge computing to reduce the cost of servers and avoid the excess energy consumption from radio transmissions. To evaluate

Table 6.1: Sensor Communication and Decision-making Methods Applied by Recent Frost Protection Systems.

Year	Decision-making Method	Sensor communication method
2008 [229]	Fuzzy logic	Unspecified wireless communication
2009 [230]	Simple conditions	Wired to a local server room
2009 [226]	Simple conditions	Wired
2012 [231]	Fuzzy logic	Unspecified wireless communication
2017 [16]	Frost index IG	Server-based computation
2019 [232]	Regressions	LoRa, RESTful
2019 [233]	Regressions	Unspecified wireless communication, cloud

this excess energy consumption, an application setting based on LoRa is also implemented for this chapter. As a summary, the major contributions of this chapter are listed below:

1. Evaluating the possibility to generalize frost prediction models from multiple locations.
2. Eliminating the requirement of centralized servers for recent frost protection systems.
3. Comparing the energy and resource consumption for LoRa-based and edge-based frost prediction systems.
4. Proposing an edge-based frost prediction system.

The rest of this chapter is organized as follows. The methodology is specified in Section 6.2. This includes the methods for model construction, details of the testing node, and settings for the experiments conducted. The experiment results are analyzed in Section 6.3 followed by the discussions on limitations. Finally, Section 6.4 summarizes the whole chapter.

6.2 Methodology

In this section, the processes involved to obtain the experiment results are outlined. First, the frost prediction models are trained and tested with data from multiple weather stations. Then, a node is assembled for environmental data collection, model inference on edge, and data transmission using LoRa. With the prediction models and testing node, a series of experiments are conducted to output results for further analysis.

6.2.1 Model Construction

There are two sets of models constructed for the experiments. Each model of the first set is constructed with environmental data collected from a single and different weather station. These

models all require local data from their corresponding weather stations. Therefore, data collection must be performed for new sites to construct these prediction models. The first set of models is used as the baseline in the experiments to represent results from existing near real-time methods. These ANN models are mentioned as "local models" or "baseline models" in the rest of the chapter.

The second set of models contains the spatially generalized models. There are only five ANN models trained with five-fold validation. Unlike each of the baseline models trained with data from a single location, the spatially generalized models are boosted by the amount of data from all the training stations in each fold. Environmental data of the weather stations in the current testing fold are used as testing data, and all other data are used as training data to construct the current model. The distribution of folds can be found in Appendix A. The purpose of five-fold validation is to test the performance of models on "new" sites. In each model or fold, the testing weather stations are not involved during training. The testing weather stations can be treated as new sites to the models. The second set of models is an attempt to test the generalization ability of neural networks on frost predictions with training and testing data from multiple locations. Therefore, the models are described as "generalized models" in the later sections.

Data for model training and testing are acquired from the Bureau of Meteorology [261]. These data are minute-wise temperature and relative humidity readings measured from 75 weather stations located in the states of NSW and ACT, Australia. Data collected in the year 2017 are used during training. Testing results of the models are obtained by model inference with data from the Winter months (June, July, August [280]) of the year 2018.

6.2.2 Testing Node

The testing node is assembled from two components. A Pycom LoPy 4 development board [302] is the major part of the testing node. This board is equipped with an Xtensa dual-core 32-bit LX6 microprocessor. The board acted as a platform to collect, process, and transmit sensor readings during experiments. The on-board Semtech SX1276 LoRa transceiver allows low-power wireless transmissions of sensor readings.

The sensing capability of the testing node is provided by a DHT22 sensor packaged as the SEN0137 module from DFRobot [23]. Temperature and relative humidity readings from the module are obtained by the Pycom LoPy 4 board to be transmitted or further processed. Table 6.2 shows the technical specifications of the sensor module.

6.2.3 Experiments

Figure 6.1 shows a high level structure of the experiments. Experiments are divided into two sets. The first set is designed to test the daily energy consumption of the testing node under different application settings. Application settings can be further split into three groups. The first group simulates the scenario that sensor readings are acquired and transmitted to a central

Table 6.2: DFRobot SEN0137 Sensor Specifications [23].

Parameter \ Sensor Type	Temperature	Humidity
Range	-40-80°C	0-100%
Resolution	0.1°C	0.1%
Error	$< \pm 0.5°C$	$\pm 2\%$

processing unit for frost prediction periodically. The device enters deep sleep mode after sensing and transmitting to save power until the next cycle. The fixed radio transmission parameters are set according to Table 6.3 Before operation with the testing node, preliminary simulations of daily LoRa transmission energy consumption is conducted using NS3 [303] based on the current consumption values of the Pycom LoPy 4 (Table 6.4).

Table 6.3: Fixed LoRa Transmission Parameters

Parameter	Value
Frequency	915.2 MHz
Transmission Power	20 dBm
Spreading Factor	12
Payload Size	4 Bytes

Table 6.4: NS3 LoRa Daily Energy Consumption Simulation Parameters

Parameter	Value
Payload Size	4 Bytes
Supply Voltage	5 V
Sleep Current	19.5 mA [302]
Transmission Current	105 mA [302]
Transmission Period	1, 5, 10, 15, 20, 30, 60 Minutes

The second group processes sensor readings locally by the testing node with a neural network to predict the next hour minimum temperature as the frost indicator. This model is the local model constructed with historical data from the site of deployment. Sensing and model inference are also triggered periodically. The final application group setting is similar to the second group setting. However, the final application group setting utilizes the generalized models created with data from multiple weather stations. Within each application group, the execution/transmission periods are also variable. Table 6.5 outlines the conditions of the experiments for daily energy consumption measurements.

The second experiment set involves simulations of the prediction models with weather station data collected in the year 2018. The loss and accuracy of the generalized models are compared

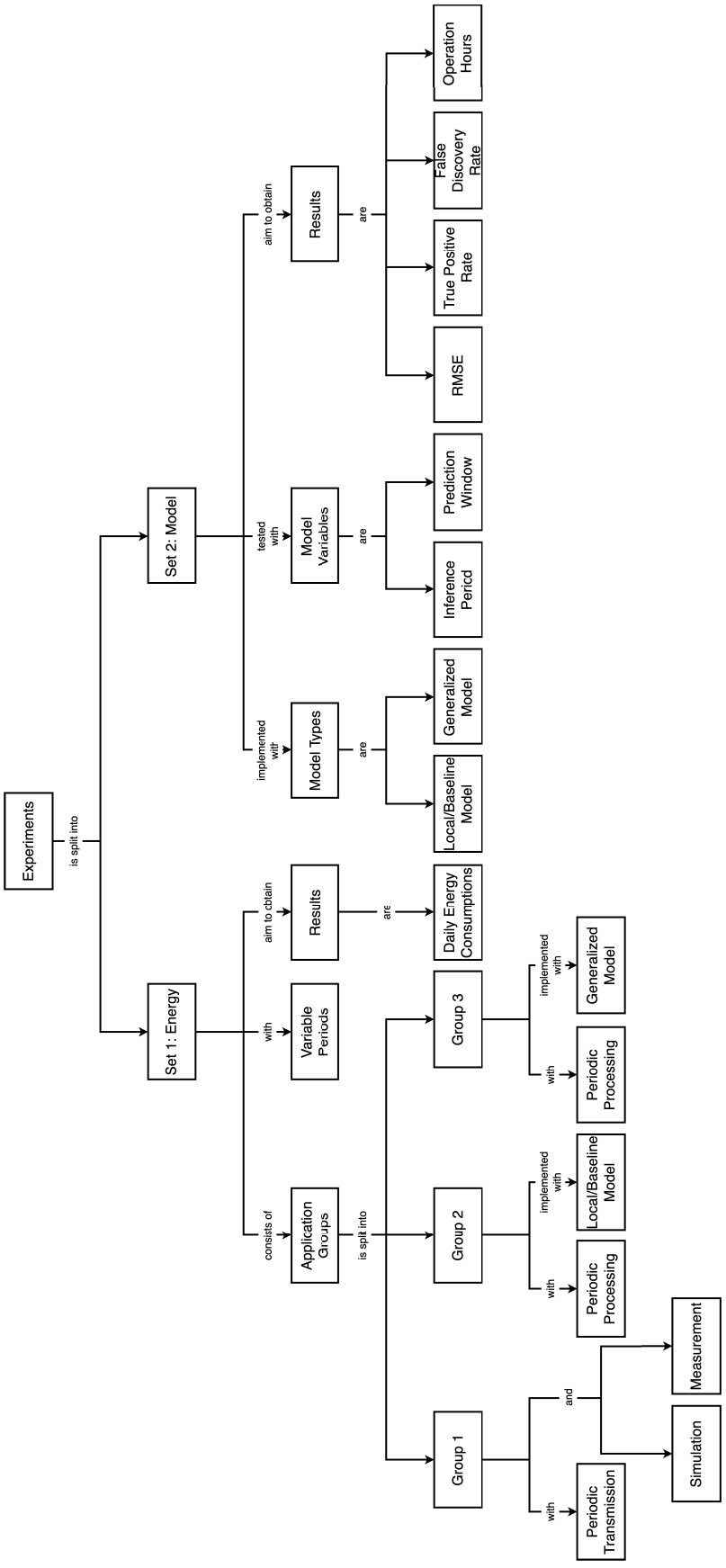


Figure 6.1: Concept Diagram of the Experiments.

Table 6.5: Daily Energy Consumption Experiments Settings

Group Number	Transmission or Processing	Model Type	Transmission/Processing Period (Minutes)
1	Transmission	N/A	1, 5, 10, 15, 20, 30, 60
2	Processing	Local	1, 5, 10, 15, 20, 30, 60
3	Processing	Generalized	1, 5, 10, 15, 20, 30, 60

with the baseline models optimized for single weather stations. After that, the true positive rates, false discovery rates, and frost protection equipment operation times are also compared between generalized and baseline models with variable transmission periods. These values are measured assuming that a frost event is below $0°C$. Transmission periods are identical to the settings of experiment set one (Tables 6.4 and 6.5).

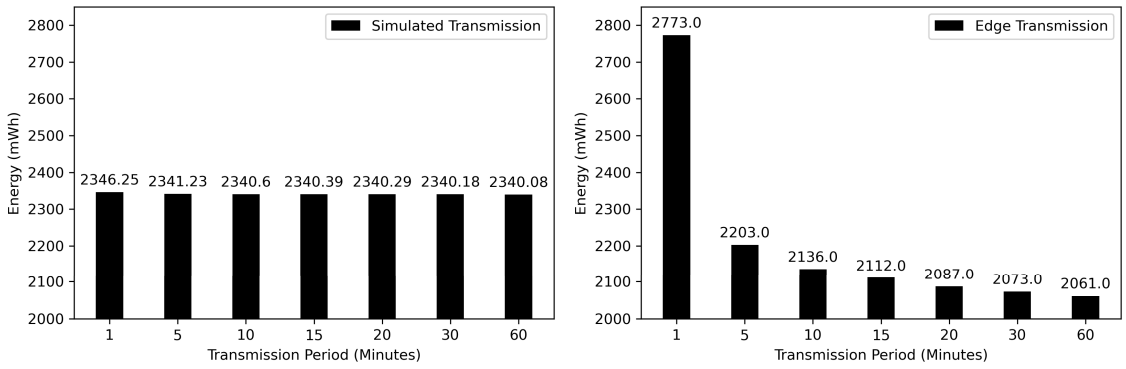
6.3 Results and Discussions

This section outlines and discusses the results of the experiments. Daily energy consumption readings acquired from experiment set one with different operation settings are discussed. After comparing the differences between the edge transmission and edge inference settings, the accuracy of frost detection for local and generalized models is evaluated with key indicators. These indicators include model loss, true positive rate, false discovery rate, and operation hour of frost protection equipment. The analysis of these results leads to the final discussions and limitations.

6.3.1 Daily Energy Consumption of the Testing Node

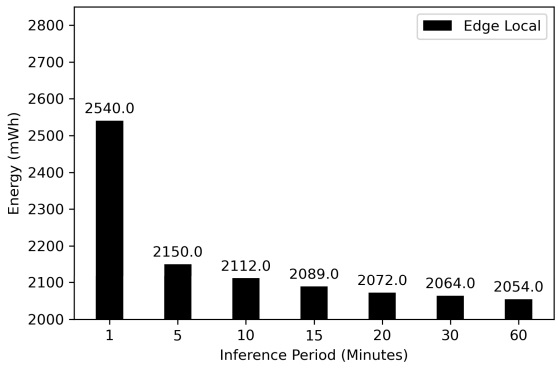
Figure 6.2 shows that the energy consumption increases with the increase of transmission/inference period in the simulated environment and all three application settings. However, the simulated result (Figure 6.2a) has significantly more minor differences between adjacent readings than the other application settings. This is due to a high current spike when the Pycom board is starting up [304] and returning from deep sleep mode (similar to reset [305]). Energy consumption by the high current spike accumulates for each transmission/inference. Therefore, as the frequency of transmission/inference increases, the effect of the current spike magnifies.

Table 6.6 compares the daily energy consumption between different application settings for different transmission/inference periods. It reveals that when the testing node is transmitting, the energy consumption readings are the highest compared with non-transmission settings at each period. However, only when the transmission period is one minute, the increase of consumption from the edge transmission is significant (greater than 5%) to the other settings. When the period increases, the energy consumption differences between the transmission and local inference settings reduce.

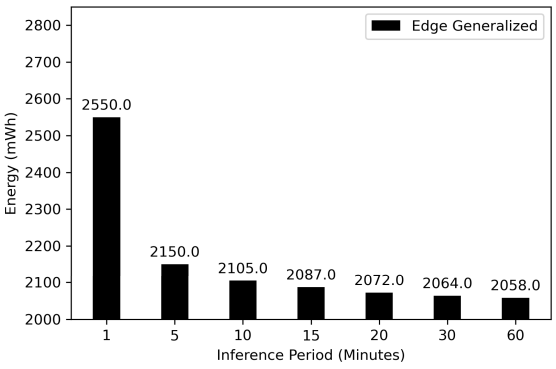


(a) Simulated Transmission

(b) Edge Transmission



(c) Edge Local



(d) Edge Generalized

Figure 6.2: Simulated and Measured Daily Energy Consumption using local or generalized models with variable transmission/inference periods.

For the latter two scenarios utilizing local frost detection with edge inference, the energy consumption readings (Table 6.6) for each inference period are similar. As the duty cycles of the two scenarios are the same, the only difference is the structure of the model deployed on the testing node. In the experiments, the structures of the models are the same with different parameters. Therefore, the energy consumption for the two application settings should be similar. The next part of the analysis is about the differences between the two models.

Table 6.6: Measured Daily Consumption (mWh) with Variable Transmission/Inference Periods (Bold values are the lowest in each column).

Period (Minutes) / **Method**	1	5	10	15	20	30	60
Edge Transmission	2773	2203	2136	2112	2087	2073	2061
Edge Local	**2540**	**2150**	2112	2089	**2072**	**2064**	**2054**
Edge Generalized	2550	**2150**	**2105**	**2087**	**2072**	**2064**	2058

6.3.2 Frost Detection with Different Model Types

Table 6.7 records the mean RMSE of the two prediction models. In general, the generalized models have lower losses than the baseline models. This is expected as each of the generalized models is trained with data from 60 weather stations, whereas every baseline model is only trained with data from one weather station. More training data ensures the generalized models can better fit the pattern between the temperature, humidity inputs, and the the next hour minimum temperature outputs. However, Table 6.8 shows that the baseline models have a slightly higher mean true positive rate than the generalized models. This indicates that the baseline models are more accurate in predicting cold, extreme temperatures. As more weather station data is used to construct the generalized models, there are significantly more amount of general temperature conditions than the cold, extreme conditions. Thus, compared to the baseline models, the generalized models fit better to the general temperature conditions than the cold, extreme conditions. This phenomenon is further explained when analyzing the false discovery rate results later in this subsection.

In Table 6.8, the true positive rate decreases with the reduction of inference frequency for both model types. As the number of inferences reduces, it also reduces the fault tolerance to predict cold temperatures. Overall, the decline in true positive rate is insignificant (smaller than 5%)) for both model types. With one inference per hour, both model types can still detect over 94% of frost events. This setting could be suitable for scenarios with a scarce energy supply.

Table 6.7: Mean RMSE of the Baseline and Generalized Models.

Model	**Mean RMSE** ($^{\circ}C$)
Baseline	0.9733
Generalized	0.8678

Table 6.8: Mean True Positive Rate of the Baseline and Generalized Models with Variable Periods (Bold values are the highest in each column).

Period (Minutes) / Model	1	5	10	15	20	30	60
Baseline	**0.9987**	**0.9985**	**0.9979**	**0.9967**	**0.9952**	**0.9906**	**0.9695**
Generalized	0.9864	0.9858	0.9847	0.9830	0.9807	0.9742	0.9475

Similar to the mean true positive rate, the mean false discovery rate also decreases with the reduction of inference frequency for both model types (Table 6.9). The generalized models have a lower false discovery rate than the baseline models for each inference period. This can be explained with the results of the better true positive rates from the baseline models. As mentioned above, the generalized models are trained with significantly more data of general temperature conditions than the extreme, cold conditions. Therefore, predicting the next hour temperature of normal temperature conditions would be more accurate and more likely to deny these conditions as frost events for generalized models to the baseline models. The results reveal the tradeoff of a small reduction of true positive rate for a significant decrease of false positives between the two model types. From an energy perspective, the generalized models would save more resources by reducing the operation time of the frost protection equipment on false positives.

Table 6.9: Mean False Discovery Rate of the Baseline and Generalized Models with Variable Periods (Bold values are the lowest in each column).

Period (Minutes) / Model	1	5	10	15	20	30	60
Baseline	0.6609	0.5810	0.5314	0.5229	0.4889	0.4587	0.3972
Generalized	**0.4416**	**0.4228**	**0.4036**	**0.3880**	**0.3739**	**0.3489**	**0.2891**

Table 6.10: Mean Operation Hours of the Protection Equipment Controlled with the Baseline and Generalized Models with Variable Periods (Bold values are the lowest in each column).

Period (Minutes) / Model	1	5	10	15	20	30	60
Baseline	227.82	184.33	164.72	161.59	150.61	141.55	124.39
Generalized	**136.62**	**132.10**	**127.71**	**124.23**	**121.14**	**115.72**	**103.09**

6.3.3 Composition of Equipment Operation Time

The mean operation hours of both model types are directly proportional to the inference frequency (Table 6.10). Suggested by Tables 6.8 and 6.9, the amount of detected true and false positives all decreases with the reduction of inference frequency. As a result, the equipment operations triggered by these positives also decline with the inference frequency. On the other hand, the

mean operation hours of the generalized models are significantly shorter than the baseline models for every period. This outcome is consistent with the similar patterns found for the mean false discovery rates.

Other than the insights provided by Table 6.10, Figure 6.3 separated the operation hours into two portions. From Figure 6.3a, the local or baseline models waste significant amounts of operation hours on false positives. Within most period settings, over 50% of operation time is contributed on false positive. Only at periods of 20, 30, and 60 minutes, the baseline models contribute more time to true positives. On the other hand, as a result of the reduced false discovery rate, the generalized models contribute more time to true positives for all period settings. Therefore, from an energy efficiency perspective, the generalized models outperform the baseline models due to shorter operation time and less time wasted on false positives in each period setting.

6.3.4 Frost Detection with Reduced Prediction Windows

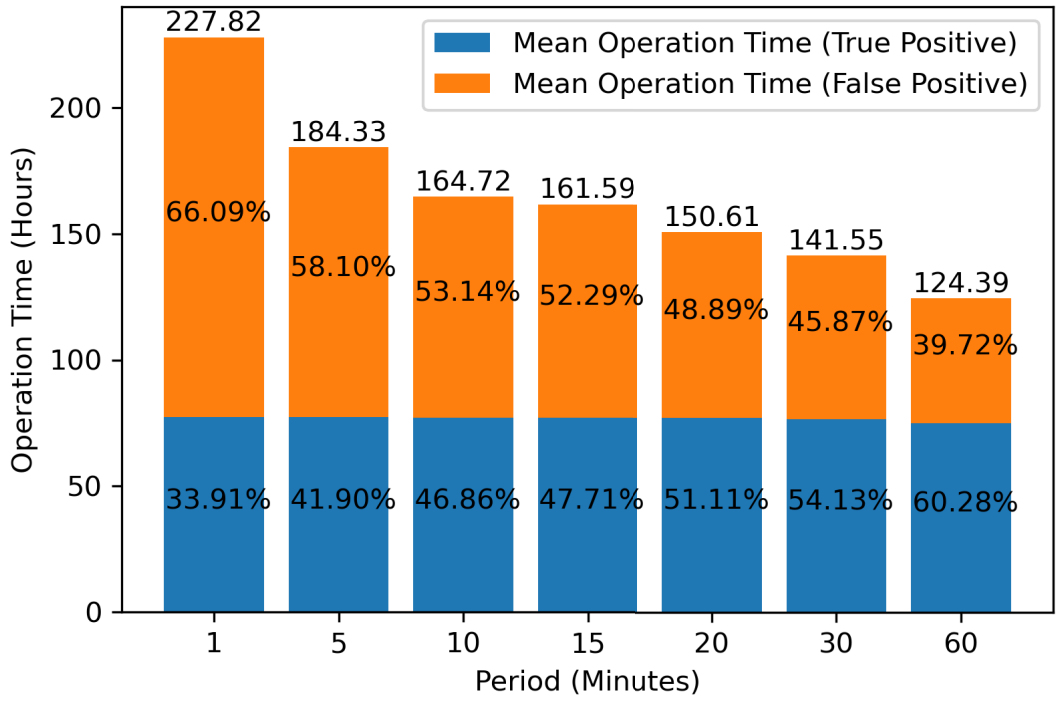
The baseline and generalized models tested in the subsections above adopted a prediction window of an hour. Hence, a possible future events could be assessed multiple times if the period of prediction is less than an hour. Multiple assessments can provide fault tolerance. However, the number of false positives can also increase with the number of assessments. This subsection evaluates new sets of baseline and generalized models with smaller prediction windows of 15 and 30 minutes. Models with prediction windows of 15 and 30 minutes are only tested with periods less and equal than 15 and 30 minutes, respectively.

Table 6.11 compares the mean RMSE of the baseline and generalized models with 15, 30, and 60 minutes prediction windows. For both model types, the loss increases with the prediction window. This is simply another demonstration that neural networks are more accurate for short-term time-series predictions than long-term time-series predictions [306]. At these shorter prediction windows, the losses of the baseline and generalized models became similar. Also, the baseline models have insignificantly smaller losses than the generalized models at these shorter windows.

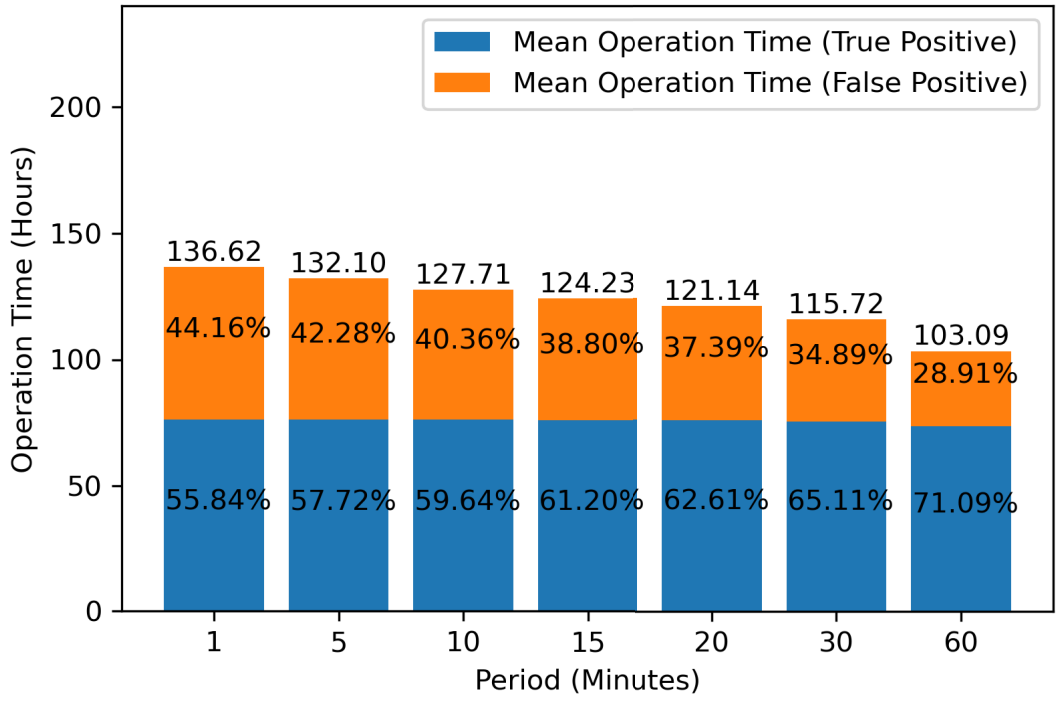
Table 6.11: Mean RMSE ($^{\circ}C$) of the Baseline and Generalized Models With Variable Prediction Windows.

Window (Minutes) / **Model**	15	30	60
Baseline	0.3760	0.5508	0.9733
Generalized	0.3793	0.5587	0.8678

The mean true positive rate decreases with the reduction of the prediction window (Table 6.12). Generally, as the prediction window is short, there are fewer assessments for a possible future event to provide extra fault tolerance. Therefore, there are fewer events captured by the models. On the other hand, fewer assessments also lead to fewer false positives. Table 6.13 reveals that with a short prediction window, the false discovery rate could be reduced down to 14.14%.



(a) Edge Local



(b) Edge Generalized

Figure 6.3: Protection Equipment Mean Operation Time Compositions of the Baseline and Generalized Models with Variable Periods.

Table 6.12: Mean True Positive Rate of the Baseline and Generalized Models with Variable Periods and Prediction Windows.

Period (Minutes) / **Model_Window (Minutes)**	1	5	10	15	20	30
Baseline_15	0.9809	0.9781	0.9682	0.9580	N/A	N/A
Baseline_30	0.9832	0.9824	0.9795	0.9753	0.9672	0.9546
Baseline_60	0.9987	0.9985	0.9979	0.9967	0.9952	0.9906
Generalized_15	0.9850	0.9819	0.9713	0.9605	N/A	N/A
Generalized_30	0.9854	0.9842	0.9804	0.9752	0.9652	0.9511
Generalized_60	0.9864	0.9858	0.9847	0.9830	0.9807	0.9742

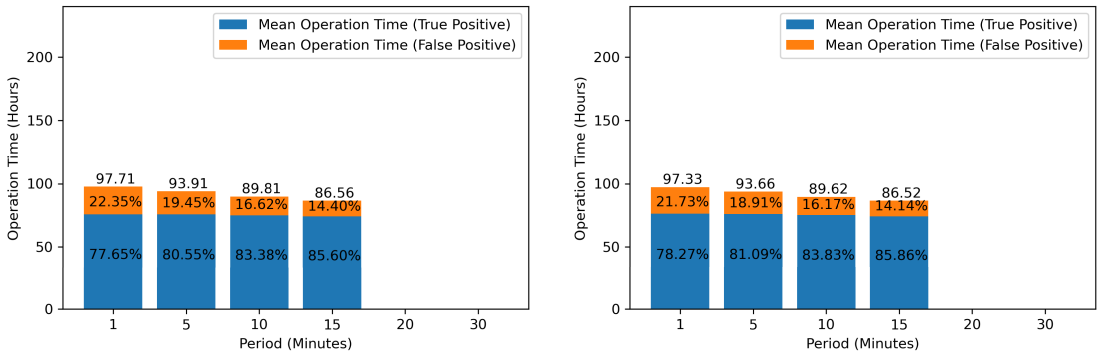
Table 6.13: Mean False Discovery Rate of the Baseline and Generalized Models with Variable Periods and Prediction Windows.

Period (Minutes) / **Model_Window (Minutes)**	1	5	10	15	20	30
Baseline_15	0.2235	0.1945	0.1662	0.1440	N/A	N/A
Baseline_30	0.3261	0.3022	0.2784	0.2594	0.2404	0.2108
Baseline_60	0.6609	0.5810	0.5314	0.5229	0.4889	0.4587
Generalized_15	0.2173	0.1891	0.1617	0.1414	N/A	N/A
Generalized_30	0.3046	0.2803	0.2566	0.2371	0.2182	0.1897
Generalized_60	0.4416	0.4228	0.4036	0.3880	0.3739	0.3489

With a reduction of false discovery rate with prediction window, Figure 6.4 shows a decline of mean operation hour with prediction window. Also, the operation time wasted on false positives is significantly reduced, especially on the baseline models. From a prediction window of 60 to 30 minutes, the time spent over false positives reduced from over and near 50% to all below 35%. When the prediction windows and periods are equal, the generalized models are still in favor with less time wasted on false positives and shorter overall operation time than the baseline models.

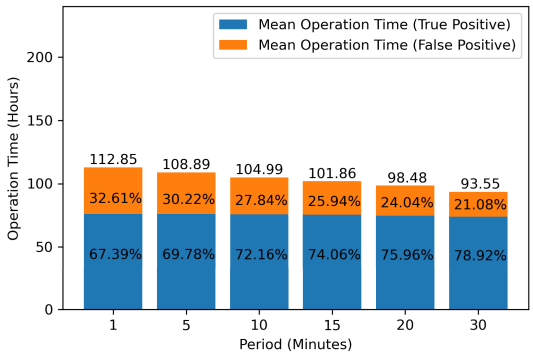
6.3.5 Limitations and Open Challenges

The first set of experiments shows that the energy consumption of local inference is less than transmission on the Pycom-based testing sensor node. However, as the frequency of transmission/inference reduces, the consumption difference becomes less significant. For the local inference scenarios, as the model structures of both local and generalized settings are the same, the daily energy consumption is similar. The accuracy of the model results seems to be more notable for affecting the operation times of the frost protection equipment. As the result of experiment set two, the generalized models have lower mean false discovery rates than the baseline models for all experimented periods. Thus, the generalized models require less operation time to achieve a

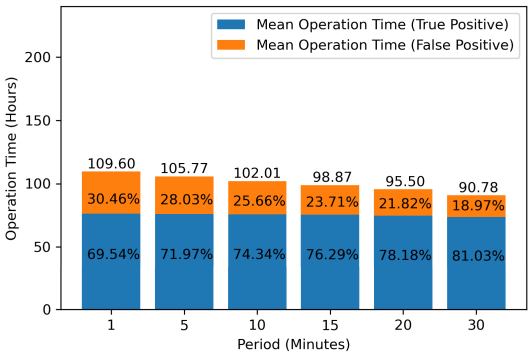


(a) Baseline_15

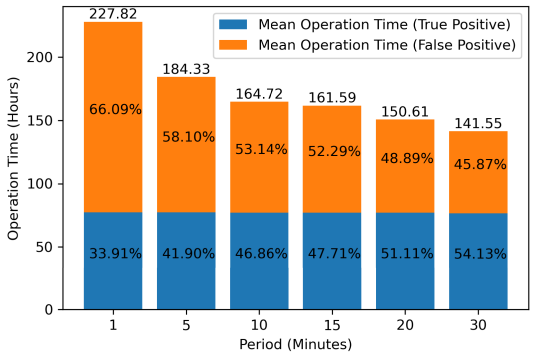
(b) Generalized_15



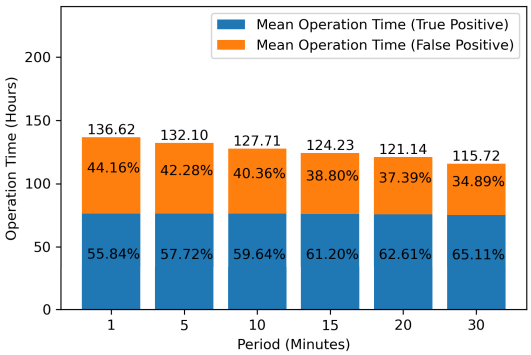
(c) Baseline_30



(d) Generalized_30



(e) Baseline_60



(f) Generalized_60

Figure 6.4: Protection Equipment Mean Operation Time Compositions of the Baseline and Generalized Models with Variable Periods and Prediction Windows.

similar true positive rate to the baseline models. Moreover, the generalized models invest most operation time on true positives, whereas the baseline models waste most operation time responding to false positives. With the above analysis of the results, some limitations are identified.

6.3.5.1 High Amount of False Positives

In the second set of experiments, the generalized models with a low prediction window improved from the baseline models with lower numbers of false positives. However, the lowest mean false discovery rate is 0.1414. This means there is over 14.14% of operation time spent on false positives. Since the results demonstrated the tradeoff between true positive rate and false discovery rate with the change of training data. In future works, more weather stations could be introduced during training to reduce the false discovery rate further.

6.3.5.2 The Effects of On-site Protection Equipment are not considered

All three application settings in this chapter assume the frost protection equipment deployed in the same site with the testing sensor node. When the sensor node provides results that trigger the protection equipment, the operation of the equipment could corrupt the readings of the sensor node. Thus, the prediction results might be corrupted. To avoid these corruptions, an external data source could be placed as a backup during the operation of the protection equipment. However, the operation area of the equipment, and the spatial differences between the external data source and the site need to be carefully investigated.

6.3.5.3 Environmental Data are not Recorded

This chapter did not mention any design to record the sensor readings after the sensor node is deployed for edge inference application settings. Consequentially, the models cannot be improved to fit new patterns of the changing climate. The two suggested solutions require excess energy. The first solution is the addition of a data storage module to the sensor node. Considering the insignificant increase in energy consumption when the transmission frequency is low, the second option is to transmit the readings to a reachable gateway. To further reduce the cost of maintaining a site to connect the gateway, sensor nodes that are able to transmit to geosynchronous Earth orbit satellites could be considered. Such data transmission does not require a local gateway. Direct communications are established between the node and satellite with a latency of two minutes [307].

6.4 Summary

Recent frost protection systems are based on wireless transmissions and centralized servers for frost prediction. This creates excess costs on server maintenance and energy consumption on

radio transmission. To avoid these costs, the proposed frost prediction method adopts the edge computing paradigm. Frost prediction results are directly obtained on the sensor node deployed with a neural network as the prediction model. The proposed prediction model is a generalized frost prediction model with training data from multiple locations. As recent prediction models require historical data for training, some time is required to collect data on new sites. Since the generalized model is trained with data from existing weather stations, data collection for new sites is unnecessary. The proposed system is implemented with a Pycom LoPy 4 board, and DHT22 sensor. The Pycom board allows transmission with LoRa, which enables the comparison of energy consumption between the proposed edge computing setting and conventional radio transmission setting. To evaluate the performance of the proposed system, two sets of experiments are conducted. As a response to **Research Question 6**, the first set of experiments compared the energy consumption of the LoRa transmission setting, edge setting with local prediction models, and edge setting with generalized prediction models. The results demonstrate that the radio transmission setting uses more energy than the edge-based settings. However, with the decrease of transmission/inference frequencies, the difference between energy consumption is reduced. Overall, the edge-based settings consume less energy. However, if the transmission frequency is low, the difference is insignificant. The second set of experiments compared the performance of the baseline and generalized models. The generalized models fit the general temperature conditions better than the extreme, cold conditions because there are more training data that represent the general temperature conditions from multiple weather stations. As a result, the generalized models have significantly lower false discovery rates than the baseline models. As the false discovery rates are reduced, lesser operation time is wasted on the false positives than the baseline. Thus, the operation time of the frost protection equipment controlled by the generalized models is also shorter than the baseline. To further reduce the operation time wasted on false positives, models with shorter prediction windows of 15 and 30 minutes are constructed. Shorter prediction windows reduced the amount of redundant assessment to a future possible event. Therefore, the number of false positives is also reduced. Overall, the generalized models with a short prediction window are in favor.

7

Conclusions and Future Work

This final chapter concludes the thesis and suggests future directions of the research. The principal aim of this research is to construct a near real-time frost prediction system with high fault tolerance. Such a system can be categorized into the smart farm applications of IoT. Therefore, a survey that summarizes the recent application and advancement of IoT is conducted. Machine learning intelligence, IoT scalability, IoT interoperability, and user friendly IoT dimensions of IoT 2.0 are leveraged to shape the design and methodology of later experiments. The implementation of these IoT concepts needs to consider the background of frost prediction and protection. Thus, a comprehensive study on existing frost prediction, protection, methods and protection systems is carried out. This study enriched the initial motivations of the thesis. A minute-wise frost prediction method based on machine learning is adopted to lift daily prediction to near real-time prediction. However, machine learning-based methods also emphasize the limitation of historical data dependence. Therefore, the implementation of a spatial interpolation-based frost prediction method is a contribution of the thesis. This method eliminates the need for local historical data and local sensors. It is further applied in the proposed frost prediction framework as a stopping mechanism to control the protection equipment. Model-based stopping mechanism is another contribution of this thesis. The usage of external weather stations as the data sources of the stopping mechanisms can prevent the corruption of output from local sources by the operation of protection equipment. The final contribution of the thesis is the edge implementation of a frost prediction system with generalized frost prediction models in consideration of IoT scalability and user friendly IoT. A more detailed summary of the chapters can be found in the next section.

7.1 Summary of Thesis Chapters

Chapter 2 is presented in two parts. The first part provides a detailed study of relevant IoT 2.0 concepts to frost prediction and protection. Both conventional and recent IoT architectures shaped the proposed frost prediction system architectures. Machine learning intelligence, IoT scalability, IoT interoperability, and user friendly IoT are the four IoT 2.0 dimensions that contributed to the system designs in this thesis. The links between the architectures, dimensions and frost systems are revealed to answer **Research Question 3**. This part of the chapter also sets the technical background of this thesis.

The second part of Chapter 2 provides a comprehensive review of existing frost prediction, protection methods, and protection systems. There are four major topics of study. These topics are frost prediction methods, frost protection methods, integrated frost prediction systems, and IoT communication protocols for frost protection applications. The study of these topics introduced the initial problems of this thesis and affected later designs of the proposed solutions. Overall, **Research Questions 1 and 2** are covered in the second part of Chapter 2 by pointing out the limitations of frost protection systems and reviewing the different prediction methods. The relevant limitations are historical data dependence, daily prediction, and low fault tolerance. These limitations bring other research questions. The limitation of daily predictions leads to **Research Question 4**, which is addressed in Chapter 3. The limitation of questionable fault tolerance is handled in Chapters 4 and 5 as responses to **Research Question 5**.

Chapter 3 presents a near real-time frost prediction method. This approach addresses **Research Question 4** and the limitations of existing methods revealed in Chapter 2. The proposed method lifts the existing method from daily predictions of the next 12–24 hours to minute-wise predictions of the next hour. The performance of RNN-based models as frost prediction methods is also evaluated. LSTM models have the lowest loss over the over tested RNN-models and the ANN baseline. However, with relatively small datasets, the accuracy of ANN models is more stable over time than RNN-based models. This leads to the adoption of ANN models in the rest of this thesis. The resolution of **Research Question 4** induced and emphasized the limitations on historical data dependence and the lack of equipment stopping conditions. These issues are resolved in Chapters 4 and 5, along with **Research Question 5**.

Dependency on local historical data is highlighted in Chapters 2 and 3 as a limitation for machine learning-based frost prediction. In Chapter 4, a spatial interpolation-based frost prediction method is proposed. This method predicts local frost with environmental parameters from external weather stations. Therefore, this method is independent of local historical data and sensors. A prediction result is calculated by aggregating the results from multiple weak predictors. Each weak predictor is trained with historical data from a different weather station. As there are multiple external sources, high fault tolerance can be achieved. The aggregation of the weak predictors reduced model loss. However, some of the predicted extreme conditions are also filtered out. To increase the stability of this method, weak predictors with higher accuracy are required. This method also reached a frost detection rate of 92.55%. As the frost detection rate of the proposed

model is still unstable and limited, this chapter is only viewed as a partial response to **Research Question 5**. In Chapter 5, this response is extended by combining the local sensor-based method and the external weather station-based method.

Based on the work of Chapters 3 and 4, a frost prediction framework is proposed in Chapter 5. The proposed framework achieved model-based control of the frost protection equipment. The framework adopted a modular design with three modules: the LPM, RPM, and MM. The LPM is the standard frost prediction model trained with local historical data and accepts local sensor readings. While the local sensor is unavailable or the frost protection equipment is operating, the RPM predicts frost with data from external weather stations. The MM manages and merges the results from the prior two modules. From the results, the Kalman filter is in favor over the particle filter as the merging algorithm to merge the prediction results due to the similar RMSEs and lower processing requirements to the particle filter. Compared with the traditional method, the proposed framework has higher true positive rates and lower false discovery rates when the transmission period is 1, 5, or 10 minutes. However, a high false discovery rate of 43.70% and the energy consumption of the LPM can be further improved. With higher true positive rates than the sole operation of an RPM, Chapter 5 complements the response to **Research Question 5** provided by Chapter 4. In Chapter 6, the limitations of high false discovery rates and energy consumption (**Research Question 6**) of the LPM are discussed.

The proposed prediction method in Chapter 6 aims to eliminate energy requirements and maintenance costs of centralized servers by adopting the edge computing paradigm. The energy consumption of the edge-based method is compared with the consumption of the transmission-based method as the response to **Research Question 6**. Frost prediction models are deployed directly on the local sensor node to trade the higher radio energy requirement for the lower computation energy requirement. Other than that, spatially generalized frost prediction models are proposed to remove the requirement of historical data for training the more common local prediction models. These generalized models are trained by existing data from multiple weather stations. Therefore, data collection is no longer required at new sites. The energy consumption of the proposed edge-based method is tested against a transmission-based method deployed with LoRa. As a result, the energy consumption of the edge-based method is less than the transmission-based method. However, this difference is reduced with the increase in inference/transmission period. As for the false discovery rate, the replacement of the baseline models with the generalized models induced a 40.03% decrease in frost protection equipment operating hours. The decline of the prediction window from an hour to 30 minutes and 15 minutes further reduced the false discovery rate to 14.14%.

7.2 Future Work

The original motivation of this research expressed concerns about possible food shortages caused by the growing world population. Frost predicting and protection systems were studied to address these concerns. From these studies, there are still limitations to be addressed shaped by the visions

of future developments. The future works derived from this thesis are:

1. Increase the accuracy of the weak predictors from the spatial interpolation-based methods.
2. Define and propose important requirements of standard datasets to enable fair comparisons between future frost prediction model types.
3. Specify a set of important and vulnerable plants for frost and study their frost tolerance. This study could later standardize the model accuracy requirements of frost prediction models.

7.2.1 Increase the Accuracy of Weak Predictors

The true positive and false discovery rates of frost detection methods are limited by the accuracy or loss of the weak predictor corresponding to each weather station. The weak predictors utilize end-to-end frost prediction methods based on spatial interpolation. Other methods should be tested to seek the improvement of accuracy. Another method could be spatially generalized models of the weak predictors. Other than end-to-end methods, predictions with multiple stages could be evaluated. For example, the environmental parameters of the target location could be obtained through spatial interpolation. A prediction with generalized frost prediction models could provide the results using these parameters.

7.2.2 Define and Propose Important Requirements of Standard Datasets

During this study, there are no standard methods or benchmarks to evaluate the performance or effectiveness of frost prediction methods. The evaluated frost prediction and protection works referred to by this thesis experimented with local, private datasets. The establishment of standard datasets and testing procedures would enable fair comparisons between frost prediction methods and reduce the workload for future model development. The standard datasets should contain data over the world. This empowers generalized frost studies and distinct studies with specified climate, location.

7.2.3 Study the Frost Tolerance of Important and Vulnerable Plants

The proposed methods demonstrated the ability of models to predict frost-related parameters. However, the accuracy requirements of these regression models are not defined. The relationship between frost damage and related parameters should be carefully tested with controlled experiments. These experiments should reveal the major parameters inducing frost damage and the effect of changing each parameter on time to frost damage. To mitigate future food shortages, food crops intolerant to frost, such as wheat, should be studied first.

BIBLIOGRAPHY

[1] M. Alrowaily and Z. Lu, "Secure edge computing in IoT systems: Review and case studies," in *Proc. 2018 IEEE/ACM Symposium on Edge Computing (SEC)*, Bellevue, WA, Oct. 2018, pp. 440–444. 15, 11, 12, 13, 41

[2] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in IoT: The challenges, and a way forward," *Journal of Network and Computer Applications*, vol. 125, pp. 251 – 279, Jan. 2019. 15, 14

[3] P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, and S. Achiche, "Design, modelling, simulation and integration of cyber physical systems: Methods and applications," *Computers in Industry*, vol. 82, pp. 273–289, Oct. 2016. 15, 17, 18

[4] R. S. Sutton and A. G. Barto, "Finite markov decision processes," in *Reinforcement learning: An introduction*. MIT press, 2018, ch. 3, pp. 47–72. 15, 13, 32, 33

[5] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1617–1634, ThirdQuarter 2014. 15, 46, 47

[6] *Network functions virtualisation (nfv): Architectural framework*, ETSI Std. RGS/NFV-002, 2014. 15, 47, 48

[7] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015. 15, 47, 48

[8] M. R. M. Kassim and A. N. Harun, "Applications of WSN in agricultural environment monitoring systems," in *Proc. 2016 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2016, pp. 344–349. 19, 17

[9] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, Jun. 2018. 19, 35, 36

[10] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for internet of things," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 8, pp. 1399–1417, Aug. 2018. 19, 35, 36

[11] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017. 19, 36, 37, 38, 45

[12] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 796–809, Jun. 2019. 19, 49, 50, 51

[13] K. B. Perry, "Basics of frost and freeze protection for horticultural crops," *HortTechnology*, vol. 8, no. 1, pp. 10 – 15, Jan. 1998. 19, 52, 53, 61, 62, 63, 69

[14] R. L. Snyder and J. d. Melo-Abreu, "Recommended methods of frost protection," in *Frost protection: fundamentals, practice and economics.* FAO, 2005, vol. 1, ch. 2, pp. 17–40. 19, 61, 62, 63

[15] E. Smith, "Cold hardiness and options for the freeze protection of Southern Highbush blueberry," *Agriculture*, vol. 9, no. 1, p. 9, Jan. 2019. 19, 61, 62, 63

[16] M. A. Guillén-Navarro, F. Pereñíguez-García, and R. Martínez-España, "IoT-based system to forecast crop frost," in *Proc. 2017 International Conference on Intelligent Environments (IE)*, Seoul, South Korea, Aug. 2017, pp. 28–35. 19, 62, 63, 65, 66, 114, 115, 116, 130, 131

[17] Y. Hu, E. A. Asante, Y. Lu, A. Mahmood, N. A. Buttar, and S. Yuan, "Review of air disturbance technology for plant frost protection," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 3, pp. 21–28, 2018. 19, 61, 63, 64

[18] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic, "High-reliability and low-latency wireless communication for internet of things: Challenges, fundamentals, and enabling technologies," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7946–7970, Oct. 2019. 19, 65, 66

[19] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of things (IoT) communication protocols: Review," in *2017 8th International Conference on Information Technology (ICIT)*, Amman, Jordan, May 2017, pp. 685–690. 19, 67

[20] M. S. Mekala and P. Viswanathan, "A survey: Smart agriculture iot with cloud computing," in *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, Vellore, India, Aug 2017, pp. 1–7. 19, 67

[21] A. Glória, F. Cercas, and N. Souto, "Comparison of communication protocols for low cost internet of things devices," in *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, Kastoria, Greece, Sep. 2017, pp. 1–6. 19, 67

[22] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of cellular lpwan technologies for iot deployment: Sigfox, lorawan, and nb-iot," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Athens, Greece, March 2018, pp. 197–202. 19, 67, 68

[23] DFRobot. (2021) DHT22 Temperature and Humidity Module. DFRobot. [Accessed: January 10, 2022]. [Online]. Available: https://wiki.dfrobot.com/DHT22_Temperature_and_humidity_module_SKU_SEN0137 21, 132, 133

[24] S. P. Long, A. Marshall-Colon, and X.-G. Zhu, "Meeting the global food demand of the future by engineering crop photosynthesis and yield potential," *Cell*, vol. 161, no. 1, pp. 56–66, Mar. 2015. 1

[25] D. H. Cobon, M. Ewai, K. Inape, and R. M. Bourke, "Food shortages are associated with droughts, floods, frosts and ENSO in Papua New Guinea," *Agricultural Systems*, vol. 145, pp. 150 – 164, Jun. 2016. 1

[26] D.-A. An-Vo, S. Mushtaq, B. Zheng, J. T. Christopher, S. C. Chapman, and K. Chenu, "Direct and indirect costs of frost in the Australian wheatbelt," *Ecological Economics*, vol. 150, pp. 122 – 136, Aug. 2018. 1

[27] S. J. Crimp, B. Zheng, N. Khimashia, D. L. Gobbett, S. Chapman, M. Howden, and N. Nicholls, "Recent changes in southern Australian frost occurrence: implications for wheat production risk," *Crop and Pasture Science*, vol. 67, no. 8, pp. 801–811, 2016. 1, 3

[28] R. L. Snyder and J. d. Melo-Abreu, "Economic importance of frost damage," in *Frost protection: fundamentals, practice and economics.* FAO, 2005, vol. 1, ch. 1, sec. 7, pp. 11–12. 1, 4

[29] H. Hänninen, "Does climatic warming increase the risk of frost damage in northern trees?" *Plant, Cell & Environment*, vol. 14, no. 5, pp. 449–454, Jun. 1991. 1, 83, 89

[30] C. K. Augspurger, "Reconstructing patterns of temperature, phenology, and frost damage over 124 years: Spring damage risk is increasing," *Ecology*, vol. 94, no. 1, pp. 41–50, Jan. 2013. 1, 83, 89

[31] T. R. Stewart, R. W. Katz, and A. H. Murphy, "Value of weather information: A descriptive study of the fruit-frost problem," *Bulletin of the American Meteorological Society*, vol. 65, no. 2, pp. 126–137, Feb. 1984. 4, 52, 53, 54, 56, 64, 65, 66, 69, 114

[32] NRAC. (2012, Sep.) Feasibility of agricultural insurance products in australia for weather-related production risks. [Online]. Available: https://www.agriculture.gov.au/sites/default/files/sitecollectiondocuments/ag-food/drought/ec/nrac/work-prog/insurance/nrac-agricultural-insurance-report.pdf 4

[33] I. Zhou, I. Makhdoom, N. Shariati, M. A. Raza, R. Keshavarz, J. Lipman, M. Abolhasan, and A. Jamalipour, "Internet of things 2.0: Concepts, applications, and future directions," *IEEE Access*, vol. 9, pp. 70 961–71 012, 2021. 8, 10

[34] I. Zhou, J. Lipman, M. Abolhasan, N. Shariati, and D. W. Lamb, "Frost monitoring cyber–physical system: A survey on prediction and active protection methods," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6514–6527, Jul. 2020. 8, 75, 76, 91, 93, 94, 113, 129, 130

[35] Y. Zhang, "Technology framework of the internet of things and its application," in *Proc. 2011 International Conference on Electrical and Control Engineering*, Bandung, Indonesia, Sep. 2011, pp. 4109–4112. 8

[36] *Overview of the Internet of things*, ITU Std. Y.4000/Y.2060, 2012. 8

[37] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*, vol. 1, 2018. 8

[38] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018. 8

[39] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1 – 10, Jun. 2017. 8, 12, 17

[40] G. Sun, V. Chang, S. Guan, M. Ramachandran, J. Li, and D. Liao, "Big data and internet of things–fusion for different services and its impacts," *Future Generation Computer Systems*, vol. 86, pp. 1368 – 1370, Sep. 2018. 8

[41] Techiexpert. (2017) What is coming in IoT 2.0? Techiexpert. [Accessed: September 16, 2020]. [Online]. Available: https://www.techiexpert.com/what-is-coming-in-iot-2-0/ 8

[42] J. Goldfein. (2019) The Internet of Things 2.0 – the technology revolution. Mercury. [Accessed: September 16, 2020]. [Online]. Available: https://mercury.one/online-business/internet-things-2-0-technology-revolution/ 8

[43] D. Litwin. (2020) Industrial IoT: How IoT has evolved and moved into the IoT 2.0 era. MarketScale. [Accessed: September 16, 2020]. [Online]. Available: https://marketscale.com/industries/industrial-iot/iot-2-0-era/ 8

[44] J. Gomez. (2020) IoT 2.0: The intelligence of things. Koombea. [Accessed: September 16, 2020]. [Online]. Available: https://www.koombea.com/blog/iot-2-0-the-intelligence-of-things/ 8

[45] J. Carter. (2017) A closer look at the Internet of Things 2.0 – and why it's inevitable. TechRadar. [Accessed: September 16, 2020]. [Online]. Available: https://www.techradar.com/news/a-closer-look-at-the-internet-of-things-20-and-why-its-inevitable 8

[46] Web Summit. (2018) IoT2.0. Youtube. [Accessed: September 16, 2020]. [Online]. Available: https://www.youtube.com/watch?v=00K0AWbMe_U 8

[47] M. Abbas. (2018) IoT 2.0: Revolutionize Internet of Things (IoT 2.0) Using Blockchain. IoTWorld. [Accessed: September 16, 2020]. [Online]. Available: https://iotworld.co/2018/01/iot-2-0-revolutionize-internet-of-things-using-blockchain/ 8

[48] Samsung. (2019) IoT 2.0: The next phase of SmartThings engagement and growth. Youtube. [Accessed: September 16, 2020]. [Online]. Available: https://www.youtube.com/watch?v=8hGkB6AQA38 8

[49] S. Nativi, A. Kotsev, P. Scudo, K. Pogorzelska, I. Vakalis, A. Dalla Benetta, and A. Perego, "IoT 2.0 and the internet of transformation," 2020. 8

[50] J. Mongay Batalla, M. Gajewski, and K. Sienkiewicz, "Concept of IoT 2.0 platform," in *Ad-hoc Networks and Wireless*, M. Garcia Pineda, J. Lloret, S. Papavassiliou, S. Ruehrup, and C. B. Westphall, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 27–34. 8, 10

[51] N. K. Narang, "Standards matters/mentor's musings on IoT 2.0: IoT coming of age," *IEEE Internet of Things Magazine*, vol. 3, no. 2, pp. 10–11, Jun. 2020. 10

[52] B. Montgomery, "Future shock: IoT benefits beyond traffic and lighting energy optimization." *IEEE Consumer Electronics Magazine*, vol. 4, no. 4, pp. 98–100, Oct. 2015. 10

[53] T. García Ferrari, A. Hinze, and J. Bowen, "An IoT for everyone: fact or fiction?" in *Proceedings of the 32nd International BCS Human Computer Interaction Conference 32*, 2018, pp. 1–5. 10

[54] J. Park, Y. Son, D. Park, J. Cho, M. Bae, M. Han, H. Lee, J. Choi, H. Kim, and S. Hwang, "IoT based distributed intelligence technology for hyper-connected space," *Electronics and Telecommunications Trends*, vol. 33, no. 1, pp. 11–19, 2018. 10

[55] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in internet of things: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1–27, Feb. 2018. 10

[56] S. Li, L. D. Xu, and S. Zhao, "5G internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1 – 9, Jun. 2018. 10, 11

[57] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN–key technology enablers for 5G networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468–2478, Nov. 2017. 10

[58] H. Ramazanali, A. Mesodiakaki, A. Vinel, and C. Verikoukis, "Survey of user association in 5G HetNets," in *Proc. 2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*, Medellin, Colombia, Nov. 2016, pp. 1–6. 10

[59] A. Morgado, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "A survey of 5G technologies: regulatory, standardization and industrial perspectives," *Digital Communications and Networks*, vol. 4, no. 2, pp. 87 – 97, Apr. 2018. 11

[60] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van, "The tactile internet: vision, recent progress, and open challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 138–145, May 2016. 11

[61] G. P. Fettweis, "5G and the future of IoT," in *Proc. ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, Lausanne, Switzerland, Sep. 2016, pp. 21–24. 11

[62] S. M. A. Oteafy and H. S. Hassanein, "Leveraging tactile internet cognizance and operation via iot and edge technologies," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 364–375, Feb. 2019. 11

[63] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, Mar. 2014. 11

[64] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *Proc. 2017 Global Internet of Things Summit (GIoTS)*, Geneva,Switzerland, Jun. 2017, pp. 1–6. 12

[65] F. S. Abkenar and A. Jamalipour, "Eba: Energy balancing algorithm for fog-iot networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6843–6849, Aug. 2019. 12

[66] F. Shirin Abkenar and A. Jamalipour, "Energy optimization in association-free fog-iot networks," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 404–412, Jun. 2020. 12

[67] F. S. Abkenar and A. Jamalipour, "A reliable data loss aware algorithm for fog-iot networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5718–5722, May 2020. 12

[68] D. Lukač, "The fourth ICT-based industrial revolution "Industry 4.0" - hmi and the case of cae/cad innovation with eplan p8," in *Proc. 2015 23rd Telecommunications Forum Telfor (TELFOR)*, Belgrade, Serbia, Nov. 2015, pp. 835–838. 12

[69] S. Trinks and C. Felden, "Edge computing architecture to support real time analytic applications : A state-of-the-art within the application area of smart factory and industry 4.0," in *Proc. 2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, Dec. 2018, pp. 2930–2939. 12

[70] L. Farhan, S. T. Shukur, A. E. Alissa, M. Alrweg, U. Raza, and R. Kharel, "A survey on the challenges and opportunities of the internet of things (IoT)," in *Proc. 2017 Eleventh International Conference on Sensing Technology (ICST)*, Sydney, NSW, Dec. 2017, pp. 1–5. 12, 33

[71] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161 – 175, Aug. 2018. 13, 18, 23, 68

[72] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," *arXiv:1803.01164 [cs]*, Sep. 2018. 13

[73] T. Hastie, R. Tibshirani, and J. Friedman, "Introduction," in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer New York, 2009, ch. 1, pp. 1–8. 13

[74] H. Rahimi, A. Zibaeenejad, and A. A. Safavi, "A novel IoT architecture based on 5G-IoT and next generation technologies," in *Proc. 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, Canada, Nov. 2018, pp. 81–88. 15, 16

[75] G. Peralta, M. Iglesias-Urkia, M. Barcelo, R. Gomez, A. Moran, and J. Bilbao, "Fog computing based efficient iot scheme for the Industry 4.0," in *Proc. 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, San Sebastian, Spain, May 2017, pp. 1–6. 15

[76] L. Carnevale, A. Celesti, A. Galletta, S. Dustdar, and M. Villari, "From the cloud to edge and iot: a smart orchestration architecture for enabling osmotic computing," in *Proc. 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Cracow, Poland, May 2018, pp. 419–424. 15

[77] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, Jan. 2018. 15, 39, 42, 43, 44

[78] R. Shahzadi, A. Niaz, M. Ali, M. Naeem, J. J. P. C. Rodrigues, F. Qamar, and S. M. Anwar, "Three tier fog networks: Enabling IoT/5G for latency sensitive applications," *China Communications*, vol. 16, no. 3, pp. 1–11, Mar. 2019. 15

[79] N. Kalatzis, M. Avgeris, D. Dechouniotis, K. Papadakis-Vlachopapadopoulos, I. Roussaki, and S. Papavassiliou, "Edge computing in IoT ecosystems for UAV-enabled early fire detection," in *Proc. 2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, Sicily, Italy, Jun. 2018, pp. 106–114. 15, 16

[80] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, "Edge computing in IoT-based manufacturing," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 103–109, Sep. 2018. 15, 16

[81] H. Guo, J. Ren, D. Zhang, Y. Zhang, and J. Hu, "A scalable and manageable IoT architecture based on transparent computing," *Journal of Parallel and Distributed Computing*, vol. 118, pp. 5 – 13, Aug. 2018. 15, 16

[82] A. Giri, S. Dutta, S. Neogy, K. Dahal, and Z. Pervez, "Internet of things (IoT): A survey on architecture, enabling technologies, applications and challenges," in *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, Liverpool, United Kingdom, 2017, pp. 7:1–7:12. 15

[83] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017. 15

[84] J. Jamaludin and J. M. Rohani, "Cyber-Physical System (CPS): State of the art," in *Proc. 2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, Quetta, Pakistan, Nov. 2018, pp. 1–5. 17, 52

[85] C. Greer, M. Burns, D. Wollman, and E. Griffor, "Cyber-physical systems and internet of things," US Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, Special Publication (NIST SP) - 1900-202, Mar. 2019. 17

[86] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, Mar. 2002. 17

[87] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, "From wireless sensor networks towards cyber physical systems," *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 397 – 413, Aug. 2011. 17

[88] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, Aug. 2008. 17

[89] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, Apr. 2015. 17, 49

[90] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, "Industrial internet of things and cyber manufacturing systems," in *Industrial Internet of Things: Cybermanufacturing Systems*, S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, Eds. Cham: Springer International Publishing, 2017, pp. 3–19. 17, 18

[91] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (iiot): An analysis framework," *Computers in Industry*, vol. 101, pp. 1 – 12, Oct. 2018. 17, 18

[92] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and IoT applications," in *Proc. 2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*, Larnaca, Cyprus, Aug. 2017, pp. 1–8. 19

[93] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2063–2079, Jun. 2018. 19

[94] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2923–2960, Fourthquarter 2018. 19, 68, 90

[95] A. S. Ali, "Support vector machine: Itself an intelligent systems," in *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, M. R. Syed and S. N. Syed, Eds. IGI Global, 2009, pp. 501–522. 19, 20, 21

[96] X. Su, X. Yan, and C.-L. Tsai, "Linear regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 3, pp. 275–294, Feb. 2012. 21

[97] T. P. Ryan, "Logistic regression," in *Modern Regression Methods*, 2nd ed. John Wiley & Sons, 2009, ch. 9, pp. 312–384. 22, 23

[98] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, May 1991. 24

[99] A. Navada, A. N. Ansari, S. Patil, and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," in *Proc. 2011 IEEE Control and System Graduate Research Colloquium*, Shah Alam, Malaysia, Jun. 2011, pp. 37–42. 24

[100] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, Feb. 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249 24

[101] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. 24

[102] P. Cichosz, "Naïve bayes classifier," in *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, M. R. Syed and S. N. Syed, Eds. John Wiley & Sons Incorporated, 2015, pp. 118–133. 25

[103] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2, pp. 131–163, Nov. 1997. 25

[104] T. N. Phyu, "Survey of classification techniques in data mining," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, Hong Kong, China, Mar. 2009, pp. 18–20. 25

[105] M. Scanagatta, A. Salmerón, and F. Stella, "A survey on bayesian network structure learning from data," *Progress in Artificial Intelligence*, May 2019. 25

[106] M. Tsagris, "Bayesian network learning with the PC algorithm: An improved and correct variation," *Applied Artificial Intelligence*, vol. 33, no. 2, pp. 101–123, Oct. 2019. 25

[107] K. Fukumizu, L. Song, and A. Gretton, "Kernel bayes' rule: Bayesian inference with positive definite kernels," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3753–3783, Jan. 2013. 25

[108] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1 – 16, Aug. 2018. 26

[109] T. A. Kerkiri and D. Konetas, "Collaborative filtering: Inference from interactive web," in *Semantic Web Personalization and Context Awareness: Management of Personal Identities and Social Networking*. IGI Global, 2011, pp. 151–163. 26

[110] I. Goodfellow, Y. Bengio, and A. Courville, "Deep feedforward networks," in *Deep learning*. MIT press, 2016, ch. 6, pp. 163–220. 26, 28

[111] ——, "Convolutional networks," in *Deep learning*. MIT press, 2016, ch. 9, pp. 321–362. 26, 28

[112] ——, "Sequence modeling: Recurrent and recursive nets," in *Deep learning*. MIT press, 2016, ch. 10, pp. 363–408. 28, 29, 75, 83, 89

[113] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555 29

[114] H. Bakırcıoğlu and T. Koçak, "Survey of random neural network applications," *European Journal of Operational Research*, vol. 126, no. 2, pp. 319 – 330, Oct. 2000. 29

[115] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1996–2018, Fourthquarter 2014. 30

[116] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer New York, 2009, ch. 14, pp. 485–585. 30, 32

[117] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, Jul. 2017. [Online]. Available: https://doi.org/10.1145/3068335 30

[118] L. Rokach, "A survey of clustering algorithms," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2010, pp. 269–298. 30

[119] G. Molenberghs and M. G. Kenward, "The expectation-maximization algorithm," in *Missing Data in Clinical Studies*. John Wiley & Sons, Ltd, 2007, ch. 8, pp. 93–104. 31

[120] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006. 31

[121] T. Howley, M. G. Madden, M.-L. O'Connell, and A. G. Ryder, "The effect of principal component analysis on machine learning accuracy with high dimensional spectral data," in *Applications and Innovations in Intelligent Systems XIII*, A. Macintosh, R. Ellis, and T. Allen, Eds. London: Springer London, 2006, pp. 209–222. 31

[122] A. Datta, S. Ghosh, and A. Ghosh, "PCA, kernel PCA and dimensionality reduction in hyperspectral images," in *Advances in Principal Component Analysis*. Springer, 2018, pp. 19–46. 31

[123] I. Borg and P. J. Groenen, "The four purposes of multidimensional scaling," in *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005, ch. 1, pp. 3–18. 31

[124] A. N. Gorban, B. Kégl, D. C. Wunsch, A. Y. Zinovyev *et al.*, *Principal manifolds for data visualization and dimension reduction*. Springer, 2008, vol. 58. 31

[125] I. Borg and P. J. Groenen, "Classical scaling," in *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005, ch. 12, pp. 261–267. 31

[126] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5 – 30, 2006, special Issue: Diffusion Maps and Wavelets. 31

[127] J. De la Porte, B. Herbst, W. Hereman, and S. Van Der Walt, "An introduction to diffusion maps," in *Proceedings of the 19th Symposium of the Pattern Recognition Association of South Africa (PRASA 2008)*, Cape Town, South Africa, 2008, pp. 15–25. 32

[128] D. Schweizer, M. Zehnder, H. Wache, H. Witschel, D. Zanatta, and M. Rodriguez, "Using consumer behavior data to reduce energy consumption in smart homes: Applying machine learning to save energy without lowering comfort of inhabitants," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Miami, FL, USA,, Dec. 2015, pp. 1123–1129. 32

[129] I. Goodfellow, Y. Bengio, and A. Courville, "Autoencoders," in *Deep learning*. MIT press, 2016, ch. 14, pp. 493–516. 32

[130] Chang Wook Ahn and R. S. Ramakrishna, "QoS provisioning dynamic connection-admission control for multimedia wireless networks using a hopfield neural network," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 1, pp. 106–117, 2004. 32

[131] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982. 32

[132] T. Kohonen, T. Huang, and M. Schroeder, *Self-Organizing Maps*, ser. Physics and astronomy online library. Springer Berlin Heidelberg, 2001. 32

[133] H. A. A. Al-Rawi, M. A. Ng, and K.-L. A. Yau, "Application of reinforcement learning to routing in distributed wireless networks: a review," *Artificial Intelligence Review*, vol. 43, no. 3, pp. 381–416, Mar. 2015. 33

[134] R. S. Sutton and A. G. Barto, "Temporal-difference learning," in *Reinforcement learning: An introduction*. MIT press, 2018, ch. 6, pp. 119–140. 33, 34

[135] ——, "Policy gradient methods," in *Reinforcement learning: An introduction*. MIT press, 2018, ch. 13, pp. 321–338. 33

[136] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *J. Mach. Learn. Res.*, vol. 4, pp. 1107–1149, Dec. 2003. 33, 34

[137] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. 34

[138] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Jan. 2019. 34

[139] M. Chincoli, S. Stavrou, and A. Liotta, "Density and transmission power in intelligent wireless sensor networks," in *Proc. 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, Limassol, Cyprus, Jun. 2018, pp. 1518–1523. 34

[140] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec. 2017. 34

[141] F. K. Shaikh and S. Zeadally, "Energy harvesting in wireless sensor networks: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041 – 1054, Mar. 2016. 34

[142] D. Ventura, D. Casado-Mansilla, J. López-de Armentia, P. Garaizar, D. López-de Ipiña, and V. Catania, "ARIIMA: A real IoT implementation of a machine-learning architecture for reducing energy consumption," in *Proc. Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, R. Hervás, S. Lee, C. Nugent, and J. Bravo, Eds. Belfast, UK: Springer International Publishing, Dec. 2014, pp. 444–451. 34

[143] M. Zehnder, H. Wache, H. Witschel, D. Zanatta, and M. Rodriguez, "Energy saving in smart homes based on consumer behavior: A case study," in *Proc. 2015 IEEE First International Smart Cities Conference (ISC2)*, Guadalajara, Mexico, Oct. 2015, pp. 1–6. 34, 35

[144] J. S. Jang, Y. L. Kim, and J. H. Park, "A study on the optimization of the uplink period using machine learning in the future IoT network," in *Proc. 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, Sydney, NSW, Mar. 2016, pp. 1–3. 34, 35

[145] S. Dharur, C. Hota, and K. Swaminathan, "Energy efficient IoT framework for smart buildings," in *Proc. 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Coimbatore,India, Feb. 2017, pp. 793–800. 34, 35

[146] W. Song, N. Feng, Y. Tian, and S. Fong, "An IoT-based smart controlling system of air conditioner for high energy efficiency," in *Proc. 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Devon, UK, Jun. 2017, pp. 442–449. 34, 35

[147] A. Javed, H. Larijani, and A. Wixted, "Improving energy consumption of a commercial building with IoT and machine learning," *IT Professional*, vol. 20, no. 5, pp. 30–38, Sep. 2018. 34, 35

[148] S. Mwanje, G. Decarreau, C. Mannweiler, M. Naseer-ul-Islam, and L. C. Schmelz, "Network management automation in 5G: Challenges and opportunities," in *Proc. 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Valencia, Spain, Sep. 2016, pp. 1–6. 36, 52

[149] B. Keshavamurthy and M. Ashraf, "Conceptual design of proactive SONs based on the big data framework for 5G cellular networks: A novel machine learning perspective facilitating a shift in the son paradigm," in *Proc. 2016 International Conference System Modeling Advancement in Research Trends (SMART)*, Moradabad, India, Nov. 2016, pp. 298–304. 36

[150] J. Ali-Tolppa, S. Kocsis, B. Schultz, L. Bodrog, and M. Kajo, "Self-healing and resilience in future 5G cognitive autonomous networks," in *Proc. 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*, Santa Fe, Argentina, Nov. 2018, pp. 1–8. 36

[151] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, Oct. 2017. 36

[152] J. Moysen and L. Giupponi, "From 4G to 5G: Self-organized network management meets machine learning," *Computer Communications*, vol. 129, pp. 248 – 268, Sep. 2018. 37

[153] R. Amiri and H. Mehrpouyan, "Self-organizing mm wave networks: A power allocation scheme based on machine learning," in *Proc. 2018 11th Global Symposium on Millimeter Waves (GSMM)*, Boulder, CO, May 2018, pp. 1–4. 37

[154] L. Le, D. Sinh, L. Tung, and B. P. Lin, "A practical model for traffic forecasting based on big data, machine-learning, and network kpis," in *Proc. 2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Las Vegas, USA, Jan. 2018, pp. 1–4. 37

[155] L. Le, B. P. Lin, L. Tung, and D. Sinh, "SDN/NFV, machine learning, and big data driven network slicing for 5G," in *Proc. 2018 IEEE 5G World Forum (5GWF)*, Silicon Valley, CA, Jul. 2018, pp. 20–25. 37

[156] W. Tong, A. Hussain, W. X. Bo, and S. Maharjan, "Artificial intelligence for vehicle-to-everything: A survey," *IEEE Access*, vol. 7, pp. 10 823–10 843, 2019. 39, 42

[157] H. Jeong, I. Jeong, H. Lee, and S. Moon, "Computation offloading for machine learning web apps in the edge server environment," in *Proc. 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria, Jul. 2018, pp. 1492–1499. 39, 42

[158] K. Portelli and C. Anagnostopoulos, "Leveraging edge computing through collaborative machine learning," in *Proc. 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Prague, Czech Republic, Aug. 2017, pp. 164–169. 39, 41, 42

[159] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, S. Chen, and P. Hou, "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 249–261, Mar. 2018. 39, 41, 42

[160] K. Kim and Y. Hong, "Autonomous network traffic control system based on intelligent edge computing," in *Proc. 2019 21st International Conference on Advanced Communication Technology (ICACT)*, PyeongChang,South Korea, Feb. 2019, pp. 164–167. 39, 41, 42

[161] Z. Feng, S. George, J. Harkes, P. Pillai, R. Klatzky, and M. Satyanarayanan, "Edge-based discovery of training data for machine learning," in *Proc. 2018 IEEE/ACM Symposium on Edge Computing (SEC)*, Bellevue, WA, Oct. 2018, pp. 145–158. 39, 41, 42, 43, 44

[162] T. Muhammed, R. Mehmood, A. Albeshri, and I. Katib, "Ubehealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities," *IEEE Access*, vol. 6, pp. 32 258–32 285, 2018. 39, 41, 42, 43, 44

[163] Nvidia. (2019) Hardware for every situation. Nvidia. [Accessed: May 28, 2019]. [Online]. Available: https://developer.nvidia.com/embedded/develop/hardware 40

[164] Intel. (2018) Product brief Intel Movidius Myriad X VPU enhanced visual intelligence at the network edge. Intel. [Accessed: May 28, 2019]. [Online]. Available: https://movidius-uploads.s3.amazonaws.com/1532110136-MyriadXVPU_ProductBrief_final_07.18.pdf 40

[165] ——. (2019) Intel neural compute stick 2. Intel. [Accessed: May 28, 2019]. [Online]. Available: https://software.intel.com/en-us/neural-compute-stick 40

[166] Google. (2019) Dev board datasheets. Google. [Accessed: May 28, 2019]. [Online]. Available: https://coral.withgoogle.com/docs/dev-board/datasheet/ 40

[167] ——. (2019) Edge TPU performance benchmarks. Google. [Accessed: May 28, 2019]. [Online]. Available: https://coral.withgoogle.com/docs/edgetpu/benchmarks/ 40

[168] ——. (2019) USB accelerator datasheet. Google. [Accessed: May 28, 2019]. [Online]. Available: https://coral.withgoogle.com/docs/accelerator/datasheet/ 40

[169] Qualcomm. (2019) Qualcomm Snapdragon 855 mobile platform product brief. Qualcomm. [Accessed: May 28, 2019]. [Online]. Available: https://www.qualcomm.com/media/documents/files/snapdragon-855-mobile-platform-product-brief.pdf 40

[170] HiSilicon. (2019) Kirin. HiSilicon. [Accessed: May 28, 2019]. [Online]. Available: http://www.hisilicon.com/en/Products/ProductList/Kirin 40

[171] Samsung. (2019) Exynos 9820 the next-level processor for the mobile future. Samsung. [Accessed: May 28, 2019]. [Online]. Available: https://www.samsung.com/semiconductor/global.semi.static/minisite/exynos/file/solution/MobileProcessor-9-Series-9820.pdf 40

[172] MediaTek. (2019) Mediatek Helio P90 product brief. MediaTek. [Accessed: May 28, 2019]. [Online]. Available: https://d86o2zu8ugzlg.cloudfront.net/mediatek-craft/documents/mediatek-helio-p90/MediaTek-Helio-P90-Product-Brief-0119.pdf 40

[173] P. Warden and D. Situnayake, *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. " O'Reilly Media, Inc.", 2019. 41, 42

[174] J. Schneible and A. Lu, "Anomaly detection on the edge," in *Proc. MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, Baltimore, MD, Oct. 2017, pp. 678–682. 43, 44

[175] J. Moon, S. Cho, S. Kum, and S. Lee, "Cloud-edge collaboration framework for iot data analytics," in *Proc. 2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, Oct. 2018, pp. 1414–1416. 43, 44

[176] F. D. Vita, D. Bruneo, A. Puliafito, G. Nardini, A. Virdis, and G. Stea, "A deep reinforcement learning approach for data migration in multi-access edge computing," in *Proc. 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*, Santa Fe, Argentina, Nov. 2018, pp. 1–8. 43, 44

[177] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for iot applications: A learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, May 2019. 43, 44

[178] T. Akiba, H. Matsukawa, H. Narimatsu, S. Eitoku, and K. Kitamura, "Device functions virtualization architecture," in *Proc. 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, Nagoya, Japan, Oct. 2017, pp. 1–2. 44, 46

[179] G. Laput, Y. Zhang, and C. Harrison, "Synthetic sensors: Towards general-purpose sensing," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver, CO, Jun. 2017, pp. 3986–3999. 44, 45, 46

[180] L. Xu, R. Collier, and G. M. P. O'Hare, "A survey of clustering techniques in wsns and consideration of the challenges of applying such to 5G IoT scenarios," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1229–1249, Oct. 2017. 45, 46

[181] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Network*, vol. 31, no. 5, pp. 96–105, Aug. 2017. 45, 46

[182] N. Dandanov, S. R. Samal, S. Bandopadhaya, V. Poulkov, K. Tonchev, and P. Koleva, "Comparison of wireless channels for antenna tilt based coverage and capacity optimization," in *Proc. 2018 Global Wireless Summit (GWS)*, Chiang Rai, Thailand, Nov. 2018, pp. 119–123. 45, 46

[183] S. Murali and A. Jamalipour, "Mobility-aware energy-efficient parent selection algorithm for low power and lossy networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2593–2601, Apr. 2019. 45

[184] S. Berger, A. Fehske, P. Zanier, I. Viering, and G. Fettweis, "Online antenna tilt-based capacity and coverage optimization," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 437–440, Aug. 2014. 45

[185] M. R. Palattella and N. Accettura, "Enabling internet of everything everywhere: LPWAN with satellite backhaul," in *Proc. 2018 Global Information Infrastructure and Networking Symposium (GIIS)*, Thessaloniki, Greece, Oct. 2018, pp. 1–5. 46

[186] D. Arellanes and K. Lau, "Analysis and classification of service interactions for the scalability of the internet of things," in *Proc. 2018 IEEE International Congress on Internet of Things (ICIOT)*, San Francisco, CA, Jul. 2018, pp. 80–87. 46

[187] M. Abolhasan, M. Abdollahi, W. Ni, A. Jamalipour, N. Shariati, and J. Lipman, "A routing framework for offloading traffic from cellular networks to sdn-based multi-hop device-to-device networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1516–1531, Dec. 2018. 46

[188] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *Journal of Network and Computer Applications*, vol. 75, pp. 138 – 155, Nov. 2016. 48

[189] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012. 49

[190] S. Ziegler, C. Crettaz, L. Ladid, S. Krco, B. Pokric, A. F. Skarmeta, A. Jara, W. Kastner, and M. Jung, "IoT6 – moving to an IPv6-based future IoT," in *The Future Internet*, A. Galis and A. Gavras, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 161–172. 49, 50

[191] K. Dickerson, R. García-Castro, P. Kostelnik, and M. Paralič, "Standards for the IoT," in *IoT Platforms, Use Cases, Privacy, and Business Models: With Hands-on Examples Based on the VICINITY Platform*, C. Zivkovic, Y. Guan, and C. Grimm, Eds. Cham: Springer International Publishing, 2021, pp. 125–147. 50

[192] S. Patidar, D. Rane, and P. Jain, "A survey paper on cloud computing," in *Proc. 2012 Second International Conference on Advanced Computing Communication Technologies*, Jan. 2012, pp. 394–398. 50, 51

[193] D. Puthal, B. P. S. Sahoo, S. Mishra, and S. Swain, "Cloud computing features, issues, and challenges: A big picture," in *Proc. 2015 International Conference on Computational Intelligence and Networks*, Odisha, India, Jan. 2015, pp. 116–123. 50, 51

[194] X. Zhang, R. Adhikari, M. Pipattanasomporn, M. Kuzlu, and S. Rahman, "Deploying iot devices to make buildings smart: Performance evaluation and deployment experience," in *Proc. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA, Dec. 2016, pp. 530–535. 51, 52

[195] E. G. Petrakis, S. Sotiriadis, T. Soultanopoulos, P. T. Renta, R. Buyya, and N. Bessis, "Internet of things as a service (iTaaS): Challenges and solutions for management of sensor data on the cloud and the fog," *Internet of Things*, vol. 3-4, pp. 156 – 174, Oct. 2018. 51

[196] N. Ramohalli and T. Adegbija, "Modular electronics for broadening non-expert participation in stem innovation: An iot perspective," in *2018 IEEE Integrated STEM Education Conference (ISEC)*, March 2018, pp. 167–174. 51

[197] A. P. Athreya and P. Tague, "Network self-organization in the internet of things," in *Proc. 2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, New Orleans, LA, Jun. 2013, pp. 25–33. 51

[198] J. Saleem, M. Hammoudeh, U. Raza, B. Adebisi, and R. Ande, "IoT standardisation: Challenges, perspectives and solution," in *Proceedings of the 2Nd International Conference on Future Networks and Distributed Systems*, Amman, Jordan, Jun. 2018, pp. 1:1–1:9. 52

[199] R. L. Snyder and J. d. Melo-Abreu, "Freeze and frost definitions," in *Frost protection: fundamentals, practice and economics.* FAO, 2005, vol. 1, ch. 1, sec. 2, pp. 2–3. 53, 100

[200] M.-L. Sutinen, R. Arora, M. Wisniewski, E. Ashworth, R. Strimbeck, and J. Palta, "Mechanisms of frost survival and freeze-damage in nature," in *Conifer Cold Hardiness*, F. J. Bigras and S. J. Colombo, Eds. Dordrecht: Springer Netherlands, 2001, pp. 89–120. 53, 119

[201] M. E. Moeletsi and M. I. Tongwane, "Spatiotemporal variation of frost within growing periods," *Advances in Meteorology*, vol. 2017, Jul. 2017. 53

[202] A. E. Baquet, A. Halter, and F. S. Conklin, "The value of frost forecasting: a Bayesian appraisal," *American Journal of Agricultural Economics*, vol. 58, no. 3, pp. 511–520, 1976. 54, 56

[203] C. Robinson and N. Mort, "A neural network solution to the problem of frost prediction," in *Proc. UKACC International Conference on Control '96 (Conf. Publ. No. 427)*, vol. 1, Exeter, UK, Sep. 1996, pp. 136–139 vol.1. 54, 56, 57

[204] P. Sallis, M. Jarur, and M. Trujillo, "Frost prediction characteristics and classification using computational neural networks," in *Advances in Neuro-Information Processing*, M. Köppen, N. Kasabov, and G. Coghill, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1211–1220. 54, 56

[205] H. Lee, J. A. Chun, H.-H. Han, and S. Kim, "Prediction of frost occurrences using statistical modeling approaches," *Advances in Meteorology*, vol. 2016, 2016. 55, 56

[206] P. Möller-Acuña, R. Ahumada-García, and J. A. Reyes-Suárez, "Machine learning for prediction of frost episodes in the Maule region of Chile," in *Ubiquitous Computing and Ambient Intelligence*, S. F. Ochoa, P. Singh, and J. Bravo, Eds. Cham: Springer International Publishing, 2017, pp. 715–720. 55, 56

[207] A. L. Diedrichs, F. Bromberg, D. Dujovne, K. Brun-Laguna, and T. Watteyne, "Prediction of frost events using machine learning and IoT sensing devices," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4589–4597, Dec. 2018. 55, 56, 57, 59, 60, 76, 90, 94

[208] M. A. Tomkowicz and A. O. Schmitt, "Frost prediction in apple orchards based upon time series models," in *Data Analysis and Applications 1*, C. H. Skiadas and J. R. Bozeman, Eds. John Wiley & Sons, Ltd, 2019, ch. 13, pp. 181–194. 55, 57

[209] A. K. Ångström, *A study of the radiation of the atmosphere: based upon observations of the nocturnal radiation during expeditions to Algeria and to California*. Smithsonian Institution, 1915, vol. 65. 57

[210] W. C. Swinbank, "Long-wave radiation from clear skies," *Quarterly Journal of the Royal Meteorological Society*, vol. 89, no. 381, pp. 339–348, Jul. 1963. 57

[211] S. B. Idso and R. D. Jackson, "Thermal radiation from the atmosphere," *Journal of Geophysical Research (1896-1977)*, vol. 74, no. 23, pp. 5397–5403, Oct. 1969. 57

[212] W. Brutsaert, "On a derivable formula for long-wave radiation from clear skies," *Water Resources Research*, vol. 11, no. 5, pp. 742–744, Oct. 1975. 57

[213] L. Ghielmi and E. Eccel, "Descriptive models and artificial neural networks for spring frost prediction in an agricultural mountain area," *Computers and Electronics in Agriculture*, vol. 54, no. 2, pp. 101 – 114, Dec. 2006. 57, 58, 60, 76, 90, 94

[214] B. J. Dempsey and M. R. Thompson, "A heat transfer model for evaluating frost action and temperature-related effects in multilayered pavement systems," *Highway Research Record*, no. 342, Jan. 1970. 57

[215] C. Robinson and N. Mort, "A neural network system for the protection of citrus crops from frost damage," *Computers and Electronics in Agriculture*, vol. 16, no. 3, pp. 177 – 187, Feb. 1997. 57, 58

[216] V. Halley, M. Eriksson, and M. Nunez, "Frost prevention and prediction of temperatures and cooling rates using GIS," *Australian Geographical Studies*, vol. 41, no. 3, pp. 287–302, Nov. 2003. 57, 58, 76, 90, 94

[217] W. Zeng, Z. Zhang, and C. Gao, "A Levenberg-Marquardt neural network model with rough set for protecting citrus from frost damage," in *Proc. 2012 Eighth International Conference on Semantics, Knowledge and Grids*, Beijing, China, Oct. 2012, pp. 193–196. 57, 59, 60, 76, 90, 94

[218] M. Fuentes, C. Campos, and S. García-Loyola, "Application of artificial neural networks to frost detection in central Chile using the next day minimum air temperature forecast," *Chilean journal of agricultural research*, vol. 78, no. 3, pp. 327–338, 2018. 57, 59, 60, 76, 90, 94

[219] L. E. Iacono, J. L. Vázquez Poletti, C. García Garino, and I. M. Llorente, "Performance models for frost prediction in public cloud infrastructures," *Computing and Informatics*, vol. 37, no. 4, pp. 815–837, 2018. 57, 59, 60, 76, 90, 94

[220] R. Stone, G. Hammer, and N. Nicholls, "Frost in northeast Australia: trends and influences of phases of the Southern Oscillation," *Journal of Climate*, vol. 9, no. 8, pp. 1896–1909, Aug. 1996. 57, 58

[221] R. L. Snyder and J. d. Melo-Abreu, "Advection frost," in *Frost protection: fundamentals, practice and economics.* FAO, 2005, vol. 1, ch. 1, sec. 4, pp. 7–8. 61, 69

[222] B. Barfield, L. Walton, and R. Lacey, "Prediction of sprinkler rates for night-time radiation frost protection," *Agricultural Meteorology*, vol. 24, pp. 1 – 9, 1981. 61, 63

[223] P. Hamer, "The heat balance of apple buds and blossoms. part ii. the water requirements for frost protection by overhead sprinkler irrigation," *Agricultural and Forest Meteorology*, vol. 37, no. 2, pp. 159 – 174, Jul. 1986. 61, 63

[224] R. Edling, R. Constantin, and W. Bourgeois, "Louisiana citrus frost protection with enclosures and microsprinklers," *Agricultural and Forest Meteorology*, vol. 60, no. 1, pp. 101 – 110, Aug. 1992. 61, 63

[225] S. Anconelli, O. Facini, V. Marletto, A. Pitacco, F. Rossi, and F. Zinoni, "Micrometeorological test of microsprinklers for frost protection of fruit orchards in Northern Italy," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 27, no. 23, pp. 1103 – 1107, 2002. 61, 63, 64

[226] D. Lamb, "Electrically heated cables protect vines from frost damage at early flowering," *Australian Journal of Grape and Wine Research*, vol. 15, no. 1, pp. 79–84, Jan. 2009. 63, 64, 65, 66, 114, 115, 130, 131

[227] Y. Hu, C. Zhao, P. Liu, A. E. Amoah, and P. Li, "Sprinkler irrigation system for tea frost protection and the application effect," *International Journal of Agricultural and Biological Engineering*, vol. 9, no. 5, pp. 17–23, 2016. 61, 63

[228] V. Beyá-Marshall, J. Herrera, F. Santibáñez, and T. Fichet, "Microclimate modification under the effect of stationary and portable wind machines," *Agricultural and Forest Meteorology*, vol. 269-270, pp. 351 – 363, May 2019. 61, 63, 64

[229] M. A. K. Jaradat, M. A. Al-Nimr, and M. N. Alhamad, "Smoke modified environment for crop frost protection: a fuzzy logic approach," *Computers and Electronics in Agriculture*, vol. 64, no. 2, pp. 104 – 110, Dec. 2008. 64, 65, 66, 114, 115, 130, 131

[230] A. A. Ghaemi, M. R. Rafiee, and A. R. Sepaskhah, "Tree-temperature monitoring for frost protection of orchards in semi-arid regions using sprinkler irrigation," *Agricultural Sciences in China*, vol. 8, no. 1, pp. 98 – 107, Jan. 2009. 64, 65, 66, 114, 115, 130, 131

[231] S. A. Alboon, A. T. Alqudah, H. R. Al-Zoubi, and A. A. Athamneh, "Fully automated smart wireless frost prediction and protection system using a fuzzy logic controller," *Int. J. Artif. Intell. Soft Comput.*, vol. 3, no. 2, pp. 165–184, Sep. 2012. 64, 65, 66, 114, 115, 130, 131

[232] J. R. Rozante, E. R. Gutierrez, P. L. da Silva Dias, A. de Almeida Fernandes, D. S. Alvim, and V. M. Silva, "Development of an index for frost prediction: Technique and validation," *Meteorological Applications*, vol. Early View, May 2019. 65, 66, 114, 115, 130, 131

[233] E. Pacini, L. Iacono, C. Mateos, and C. García Garino, "A bio-inspired datacenter selection scheduler for federated clouds and its application to frost prediction," *Journal of Network and Systems Management*, vol. 27, no. 3, pp. 688–729, Jul. 2019. 65, 66, 130, 131

[234] X. Shi, X. An, Q. Zhao, H. Liu, L. Xia, X. Sun, and Y. Guo, "State-of-the-art internet of things in protected agriculture," *Sensors*, vol. 19, no. 8, p. 1833, Apr. 2019. 66

[235] A. Ikpehai, B. Adebisi, K. M. Rabie, K. Anoh, R. E. Ande, M. Hammoudeh, H. Gacanin, and U. M. Mbanaso, "Low-power wide area network technologies for internet-of-things: A comparative review," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2225–2240, Apr. 2019. 66

[236] N. Kaur and S. K. Sood, "An energy-efficient architecture for the internet of things (IoT)," *IEEE Systems Journal*, vol. 11, no. 2, pp. 796–805, June 2017. 66, 67

[237] Eurostat, *Agriculture, forestry and fishery statistics*. Publications Office of the European Union, 2018. 66

[238] A. B. of Statistics. (2018) Agricultural commodities, australia, 2015-16. [Online]. Available: https://www.abs.gov.au/AUSSTATS/abs@.nsf/Lookup/7121.0Main+Features12015-16 66

[239] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Data collection and wireless communication in internet of things (IoT) using economic analysis and pricing models: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2546–2590, Fourthquarter 2016. 66, 67

[240] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, "Data cleaning: Overview and emerging challenges," in *Proceedings of the 2016 International Conference on Management of Data*, ser. SIGMOD '16. New York, NY, USA: ACM, Jun. 2016, pp. 2201–2206. 68

[241] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 393–430, Firstquarter 2019. 68

[242] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70 – 90, Apr. 2018. 68

[243] J. Rhee and J. Im, "Meteorological drought forecasting for ungauged areas based on machine learning: Using long-range climate forecast and remote sensing data," *Agricultural and Forest Meteorology*, vol. 237-238, pp. 105 – 122, May 2017. 69

[244] B. Wowk and G. M. Fahy, "Inhibition of bacterial ice nucleation by polyglycerol polymers," *Cryobiology*, vol. 44, no. 1, pp. 14 – 23, Feb. 2002. 69

[245] M. P. Fuller, F. Hamed, M. Wisniewski, and D. M. Glenn, "Protection of plants from frost using hydrophobic particle film and acrylic polymer," *Annals of Applied Biology*, vol. 143, no. 1, pp. 93–98, Apr. 2003. 69

[246] S. Chouikhi, I. E. Korbi, Y. Ghamri-Doudane, and L. A. Saidane, "A survey on fault tolerance in small and large scale wireless sensor networks," *Computer Communications*, vol. 69, pp. 22 – 37, Sep. 2015. 70

[247] G. Kakamanshadi, S. Gupta, and S. Singh, "A survey on fault tolerance techniques in wireless sensor networks," in *Proc. 2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, Noida, India, Oct. 2015, pp. 168–173. 70

[248] N. Verma and D. Singh, "Data redundancy implications in wireless sensor networks," *Procedia Computer Science*, vol. 132, pp. 1210 – 1217, Jun. 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918307683 70

[249] N. Vlajic and D. Xia, "Wireless sensor networks: to cluster or not to cluster?" in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM'06)*, Buffalo-Niagara Falls, NY, USA, June 2006, pp. 9 pp.–268. 70

[250] Écio Souza Diniz, A. S. Lorenzon, N. L. M. de Castro, G. E. Marcatti, O. P. dos Santos, J. C. de Deus Júnior, R. B. L. Cavalcante, E. I. Fernandes-Filho, and C. Hummeldo Amaral, "Forecasting frost risk in forest plantations by the combination of spatial data and machine learning algorithms," *Agricultural and Forest Meteorology*, vol. 306, p. 108450, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168192321001337 71, 72, 73

[251] J. Xu, S. Guga, G. Rong, D. Riao, X. Liu, K. Li, and J. Zhang, "Estimation of frost hazard for tea tree in zhejiang province based on machine learning," *Agriculture*, vol. 11, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2077-0472/11/7/607 71, 73

[252] I. Noh, H.-W. Doh, S.-O. Kim, S.-H. Kim, S. Shin, and S.-J. Lee, "Machine learning-based hourly frost-prediction system optimized for orchards using automatic weather station and digital camera image data," *Atmosphere*, vol. 12, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2073-4433/12/7/846 71, 73

[253] S. K. Bal, R. Dhakar, P. V. Kumar, A. Mishra, V. Pramod, M. Chandran, V. Sandeep, A. Rao, K. Gill, and R. Prasad, "Temporal trends in frost occurrence and their prediction models using multivariate statistical techniques for two diverse locations of northern india," *Theoretical and Applied Climatology*, vol. 146, no. 3, pp. 1097–1110, 2021. 72, 73

[254] H. Lira, L. Martí, and N. Sanchez-Pi, "A graph neural network with spatio-temporal attention for multi-sources time series data: An application to frost forecast," *Sensors*, vol. 22, no. 4, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/4/1486 72, 73, 74

[255] D. L. Gobbett, U. Nidumolu, H. Jin, P. Hayman, and J. Gallant, "Minimum temperature mapping augments australian grain farmers' knowledge of frost," *Agricultural and Forest Meteorology*, vol. 304-305, p. 108422, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168192321001052 72, 74

[256] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *arXiv preprint arXiv:1801.01078*, 2017. 75, 83, 89

[257] I. Zhou, J. Lipman, M. Abolhasan, and N. Shariati, "Minute-wise frost prediction: An approach of recurrent neural networks," *Array*, vol. 14, p. 100158, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590005622000248 75, 93, 113, 116, 117, 130

[258] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of iot in agriculture for the implementation of smart farming," *IEEE Access*, vol. 7, pp. 156 237–156 271, 2019. 75

[259] V. K. Sarker, T. N. Gia, H. Tenhunen, and T. Westerlund, "Lightweight security algorithms for resource-constrained iot-based sensor nodes," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 1–7. 75

[260] D. di Francescantonio, M. Villagra, G. Goldstein, and P. I. Campanello, "Drought and frost resistance vary between evergreen and deciduous Atlantic Forest canopy trees," *Functional Plant Biology*, Jun. 2020. 76, 89, 94

[261] Bureau of Meteorology, *Weather Station Directory*, Bureau of Meteorology, 2020, (Accessed: April 04, 2020). [Online]. Available: http://www.bom.gov.au/climate/data/stations/ 77, 95, 119, 132

[262] ——, *Australia in winter 2016*, Bureau of Meteorology, 2016, (Accessed: April 04, 2020). [Online]. Available: http://www.bom.gov.au/climate/current/season/aus/archive/201608.summary.shtml 77

[263] ——, *Australia in winter 2017*, Bureau of Meteorology, 2017, (Accessed: April 04, 2020). [Online]. Available: http://www.bom.gov.au/climate/current/season/aus/archive/201708.summary.shtml 77

[264] M. Lee, S. Moon, Y. Kim, and B. Moon, "Correcting abnormalities in meteorological data by machine learning," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, San Diego, CA, USA, 2014, pp. 888–893. 79, 96

[265] Bureau of Meteorology, *Wind*, Bureau of Meteorology, 2020, (Accessed: April 04, 2020). [Online]. Available: http://www.bom.gov.au/marine/knowledge-centre/reference/wind.shtml 79

[266] J. W. Osborne, "Six: Dealing with missing or incomplete data: Debunking the myth of emptiness," in *Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data*. SAGE Publications, Inc, 2013, ch. 6, pp. 105–138. 81

[267] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2017. 82

[268] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014. 83

[269] Google, *tf.keras.layers.LSTM*, Google, 2020, (Accessed: October 12, 2020). [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM 87

[270] ——, *tf.keras.layers.GRU*, Google, 2020, (Accessed: October 12, 2020). [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU 87

[271] S. Muzaffar and A. Afshari, "Short-term load forecasts using LSTM networks," *Energy Procedia*, vol. 158, pp. 2922 – 2927, Feb. 2019, innovative Solutions for Energy Transitions. 89

[272] K. Barlow, B. Christy, G. O'Leary, P. Riffkin, and J. Nuttall, "Simulating the impact of extreme heat and frost events on wheat crop production: A review," *Field Crops Research*, vol. 171, pp. 109 – 119, Feb. 2015. 89

[273] G. Deepika and P. Rajapirian, "Wireless sensor network in precision agriculture: A survey," in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, Pudukkottai, India, 2016, pp. 1–4. 90

[274] B. Martinez, M. Montón, I. Vilajosana, and J. D. Prades, "The power of models: Modeling power consumption for IoT devices," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5777–5789, Oct. 2015. 90, 126

[275] J. Li and A. D. Heap, "Spatial interpolation methods applied in the environmental sciences: A review," *Environmental Modelling & Software*, vol. 53, pp. 173–189, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364815213003113 93, 101

[276] T. Appelhans, E. Mwangomo, D. R. Hardy, A. Hemp, and T. Nauss, "Evaluating machine learning approaches for the interpolation of monthly air temperature at Mt. Kilimanjaro, Tanzania," *Spatial Statistics*, vol. 14, pp. 91–113, 2015, spatial and Spatio-Temporal Models for Interpolating Climatic and Meteorological Data. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211675315000482 93

[277] T. N. Carlson and D. A. Ripley, "On the relation between NDVI, fractional vegetation cover, and leaf area index," *Remote Sensing of Environment*, vol. 62, no. 3, pp. 241–252, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425797001041 94

[278] J. Wang, K. P. Price, and P. M. Rich, "Spatial patterns of ndvi in response to precipitation and temperature in the central great plains," *International Journal of Remote Sensing*, vol. 22, no. 18, pp. 3827–3844, 2001. [Online]. Available: https://doi.org/10.1080/01431160010007033 94

[279] J. Wang, P. M. Rich, and K. P. Price, "Temporal responses of ndvi to precipitation and temperature in the central great plains, usa," *International Journal of Remote Sensing*, vol. 24, no. 11, pp. 2345–2364, 2003. [Online]. Available: https://doi.org/10.1080/01431160210154812 94

[280] Bureau of Meteorology. (2018) Australia in winter 2018. Bureau of Meteorology. [Accessed: April 04, 2021]. [Online]. Available: http://www.bom.gov.au/climate/current/season/aus/archive/201808.summary.shtml 95, 132

[281] J. Gallant, N. Wilson, T. Dowling, A. Read, and C. Inskeep. (2011) SRTM-derived 1 Second Digital Elevation Models Version 1.0. Geoscience Australia. [Accessed: April 04, 2021]. [Online]. Available: http://pid.geoscience.gov.au/dataset/ga/72759 95, 119

[282] K. Didan. (2021) MOD13Q1 MODIS/Terra Vegetation Indices 16-Day L3 Global 250m SIN Grid V006. NASA EOSDIS Land Processes DAAC. [Accessed: April 04, 2021]. [Online]. Available: https://doi.org/10.5067/MODIS/MOD13Q1.006 95, 119

[283] Department of Industry, Science, Energy and Resources. (2020) NSW State Boundary - Geoscape Administrative Boundaries. Department of Industry, Science, Energy and Resources. [Accessed: April 04, 2021]. [Online]. Available: https://data.gov.au/data/dataset/a1b278b1-59ef-4dea-8468-50eb09967f18 96, 119

[284] ——. (2020) ACT State Boundary - Geoscape Administrative Boundaries. Department of Industry, Science, Energy and Resources. [Accessed: April 04, 2021]. [Online]. Available: https://data.gov.au/data/dataset/83468f0c-313d-4354-9592-289554eb2dc9 96, 119

[285] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4. 99

[286] D. R. Roberts, V. Bahn, S. Ciuti, M. S. Boyce, J. Elith, G. Guillera-Arroita, S. Hauenstein, J. J. Lahoz-Monfort, B. Schröder, W. Thuiller, D. I. Warton, B. A. Wintle, F. Hartig, and C. F. Dormann, "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure," *Ecography*, vol. 40, no. 8, pp. 913–929, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/ecog.02881 109

[287] J. Li, A. D. Heap, A. Potter, and J. J. Daniell, "Application of machine learning methods to spatial interpolation of environmental variables," *Environmental Modelling & Software*, vol. 26, no. 12, pp. 1647–1659, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364815211001654 110

[288] K. S. Bakar, P. Kokic, and H. Jin, "A spatiodynamic model for assessing frost risk in south-eastern australia," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 64, no. 5, pp. 755–778, 2015. 110

[289] ——, "Hierarchical spatially varying coefficient and temporal dynamic process models using sptdyn," *Journal of Statistical Computation and Simulation*, vol. 86, no. 4, pp. 820–840, 2016. 110

[290] O. Ghorbanzadeh, K. Valizadeh Kamran, T. Blaschke, J. Aryal, A. Naboureh, J. Einali, and J. Bian, "Spatial prediction of wildfire susceptibility using field survey gps data and machine learning approaches," *Fire*, vol. 2, no. 3, 2019. [Online]. Available: https://www.mdpi.com/2571-6255/2/3/43 110

[291] P. Ryan, N. McKenzie, D. O'Connell, A. Loughhead, P. Leppert, D. Jacquier, and L. Ashton, "Integrating forest soils information across scales: spatial prediction of soil properties under australian forests," *Forest Ecology and Management*, vol. 138, no. 1, pp. 139–157, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378112700003935 110

[292] N. L. and M. H.S., "Atmospheric weather prediction using various machine learning techniques: A survey," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 422–428. 110

[293] I. Zhou, J. Lipman, M. Abolhasan, and N. Shariati, "Intelligent spatial interpolation-based frost prediction methodology using artificial neural networks with limited local data," *Environmental Modelling & Software*, p. 105724, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136481522300110X 113, 116, 117, 122, 126

[294] Geoscience Australia. (2021) Geodetic Calculators. Geoscience Australia. [Accessed: April 04, 2021]. [Online]. Available: https://geodesyapps.ga.gov.au/sunrise 118

[295] S. Dubuisson, *Tracking with particle filter for high-dimensional observation and state spaces*. John Wiley & Sons, 2015. 120

[296] M. Van der Loo and E. De Jonge, *Statistical data cleaning with applications in R*. John Wiley & Sons, 2018. 126

[297] A. Hassan, A. Anter, and M. Kayed, "A survey on extending the lifetime for wireless sensor networks in real-time applications," *International Journal of Wireless Information Networks*, vol. 28, no. 1, pp. 77–103, 2021. 129

[298] A. Zourmand, A. L. Kun Hing, C. Wai Hung, and M. AbdulRehman, "Internet of things (iot) using lora technology," in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 2019, pp. 324–330. 129

[299] A. Lavric and V. Popa, "Internet of Things and LoRa™ Low-Power Wide-Area Networks: A survey," in *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, 2017, pp. 1–5. 129

[300] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1275–1284, 2018. 130

[301] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018. 130

[302] Pycom Ltd. (2020) Lopy 4 Datasheet. Pycom Ltd. [Accessed: January 10, 2022]. [Online]. Available: https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_LoPy4_v2.pdf 132, 133

[303] D. Magrin, M. Capuzzo, S. Romagnolo, and M. Luvisotto. (2021) LoRaWAN An ns-3 module for simulation of LoRaWAN networks. [Accessed: October 10, 2021]. [Online]. Available: https://apps.nsnam.org/app/lorawan/ 133

[304] S. Ould and N. S. Bennett, "Energy performance analysis and modelling of lora prototyping boards," *Sensors*, vol. 21, no. 23, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/23/7992 135

[305] Pycom Ltd. (2021) Firmware & API Reference: machine. Pycom Ltd. [Accessed: January 10, 2022]. [Online]. Available: https://docs.pycom.io/firmwareapi/pycom/machine/ 135

[306] H. H. Nguyen and C. W. Chan, "Multiple neural networks for a long term time series forecast," *Neural Computing & Applications*, vol. 13, no. 1, pp. 90–98, 2004. 139

[307] M. Parr. (2019) IoT + geosatellites... a perfect match. Satnews. [Accessed: September 16, 2020]. [Online]. Available: http://www.satmagazine.com/story.php?number=1830210392143

APPENDIX A
WEATHER STATION LOCATION AND FOLD DISTRIBUTION

Table A.1: Weather Station Location and Fold Distribution (Bold Stations Appeared in Chapter 3)

Station ID	**Latitude (Degrees)**	**Longitude (Degrees)**	**Fold Number**
70351	-35.3088	149.2004	0
70217	-36.2939	148.9725	0
65103	-33.3627	147.9205	0
66194	-33.9057	151.1134	0
68192	-34.0390	150.6890	0
75041	-34.2487	146.0695	0
66037	-33.9465	151.1731	0
63291	-33.4119	149.6540	0
73138	-34.2493	148.2475	0
51049	-31.9861	147.9489	0
62100	-32.7244	150.2290	0
58198	-28.8353	153.5585	0
67113	-33.7195	150.6783	0
61078	-32.7939	151.8364	0
61363	-32.0335	150.8264	0
69148	-35.9004	150.1437	1
61287	-32.1852	150.1737	1
51161	-30.9776	148.3798	1
74148	-34.7050	146.5140	1
58208	-28.8824	153.0618	1
66161	-33.9925	150.9489	1
62101	-32.5628	149.6149	1
47048	-32.0012	141.4694	1
65068	-33.1281	148.2428	1
69139	-36.6722	149.8191	1

Table A.2: Weather Station Location and Fold Distribution (Bold Stations Appeared in Chapter 3)(Cont.)

Station ID	**Latitude (Degrees)**	**Longitude (Degrees)**	**Fold Number**
59007	-31.0711	152.7717	1
58214	-28.8305	153.2601	1
60141	-31.8895	152.5120	1
68262	-34.3335	150.2670	1
66137	-33.9176	150.9837	1
58012	-29.4325	153.3632	2
75019	-34.5412	144.8345	2
56238	-30.5273	151.6158	2
63292	-33.6185	150.2741	2
49000	-32.8833	144.3092	2
67105	-33.6004	150.7761	2
63303	-33.3768	149.1263	2
58077	-29.6224	152.9605	2
68257	-34.0615	150.7735	2
66212	-33.8338	151.0718	2
55202	-30.9537	150.2494	2
68242	-34.6532	150.8609	2
74258	-35.5575	144.9458	2
65111	-33.8382	148.6540	2
58212	-29.1830	153.3964	2
70330	-34.8085	149.7311	3
48245	-30.0362	145.9521	3
54038	-30.3154	149.8302	3
72160	-36.0690	146.9509	3
72162	-36.2304	148.1405	3

Table A.3: Weather Station Location and Fold Distribution (Bold Stations Appeared in Chapter 3)(Cont.)

Station ID	**Latitude (Degrees)**	**Longitude (Degrees)**	**Fold Number**
72161	-35.9371	148.3779	3
50017	-33.9382	147.1962	3
60139	-31.4335	152.8655	3
61375	-33.2894	151.2107	3
68072	-34.9469	150.5353	3
68239	-34.5253	150.4217	3
61425	-33.4351	151.3614	3
46012	-31.5194	143.3850	3
64017	-31.3330	149.2699	3
69128	-35.1103	150.0826	3
68228	-34.3691	150.9291	4
67108	-33.8969	150.7281	4
69137	-37.2622	150.0504	4
52088	-30.0372	148.1223	4
61392	-31.7416	150.7937	4
67119	-33.8510	150.8567	4
55325	-31.0742	150.8362	4
61055	-32.9184	151.7985	4
50137	-33.0682	147.2133	4
69138	-35.3635	150.4828	4
61366	-33.2814	151.5766	4
65070	-32.2206	148.5753	4
61260	-32.7886	151.3377	4
69147	-36.9077	149.8989	4
68241	-34.5638	150.7900	4

APPENDIX B
RASTER MAPS GENERATED FROM MODELS CREATED BY DIFFERENT WEATHER STATIONS

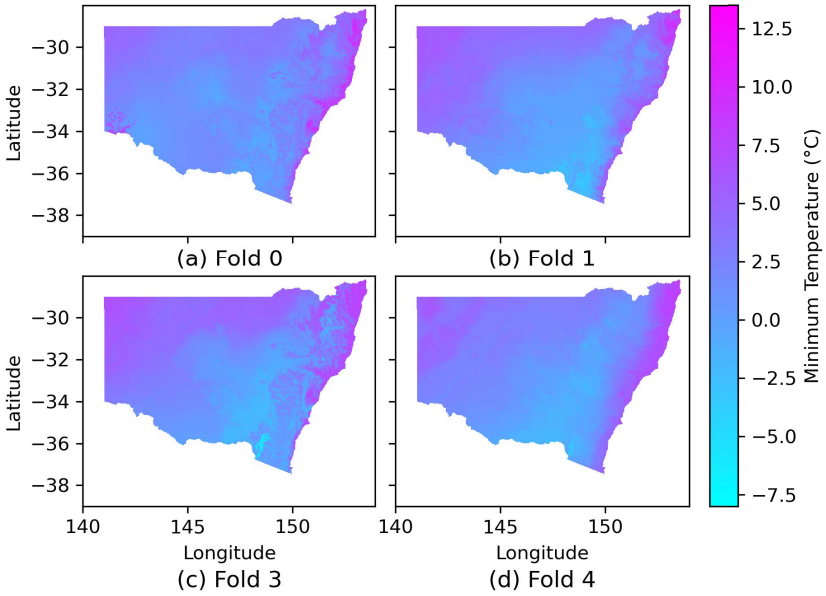


Figure B.1: Raster Maps from Models Using Weather Station 58212 as Climate Data Source Trained with the Datasets from Folds 0, 1, 3, and 4.

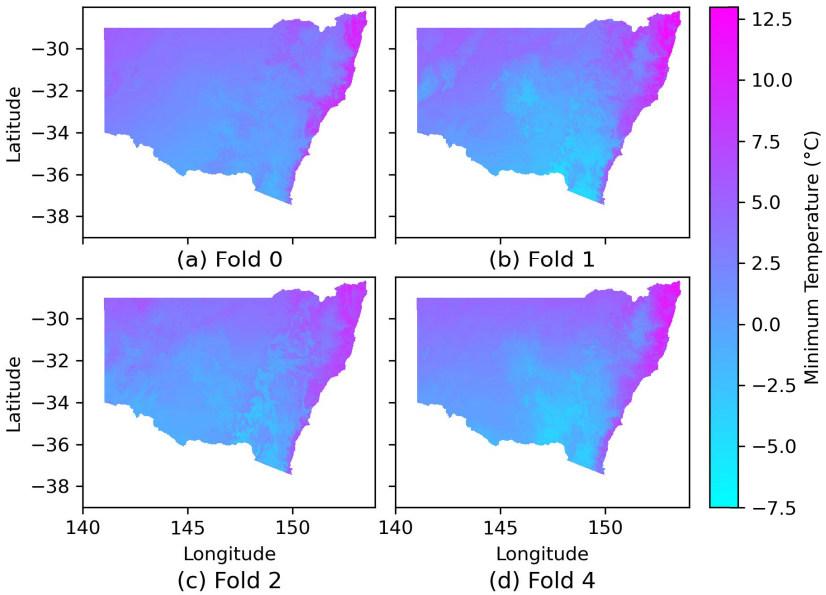


Figure B.2: Raster Maps from Models Using Weather Station 72160 as Climate Data Source Trained with the Datasets from Folds 0, 1, 2, and 4.

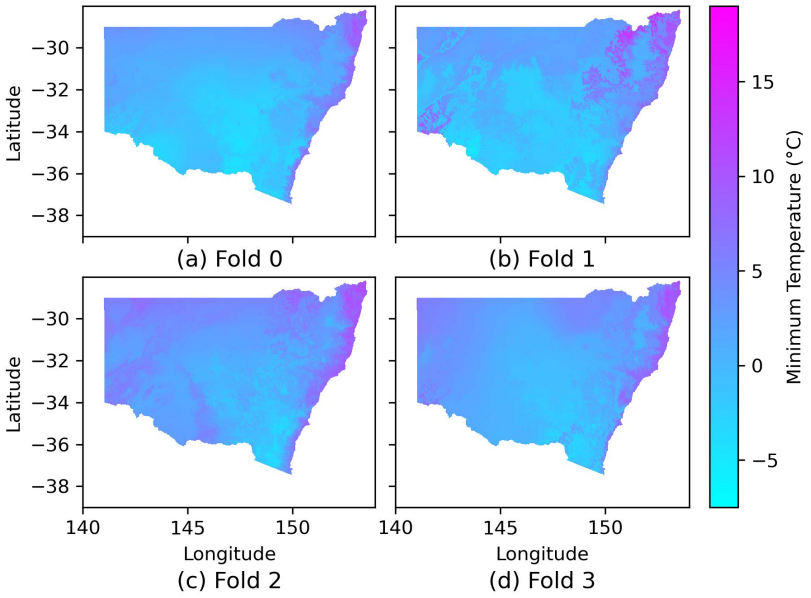


Figure B.3: Raster Maps from Models Using Weather Station 67119 as Climate Data Source Trained with the Datasets from Folds 0–3.

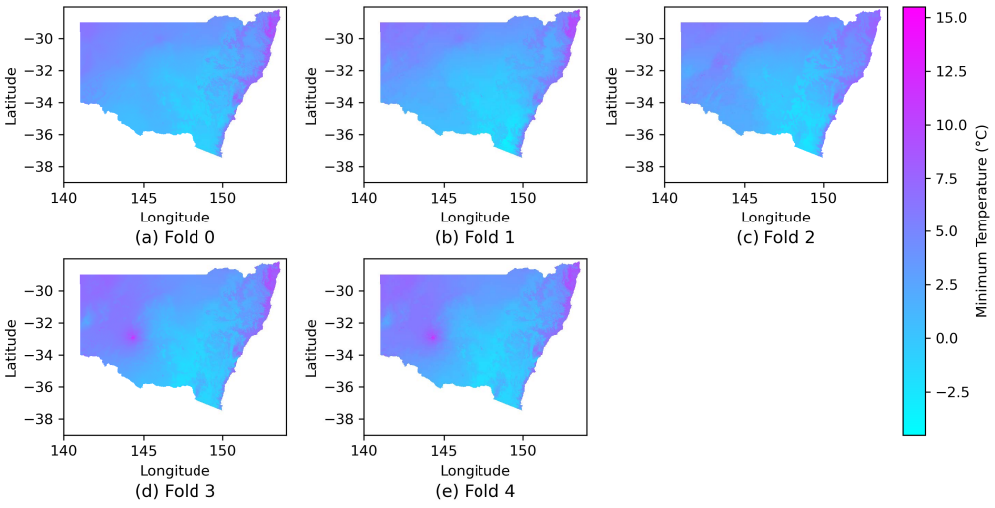


Figure B.4: Raster Maps from Weighted Averaged Results per Fold.

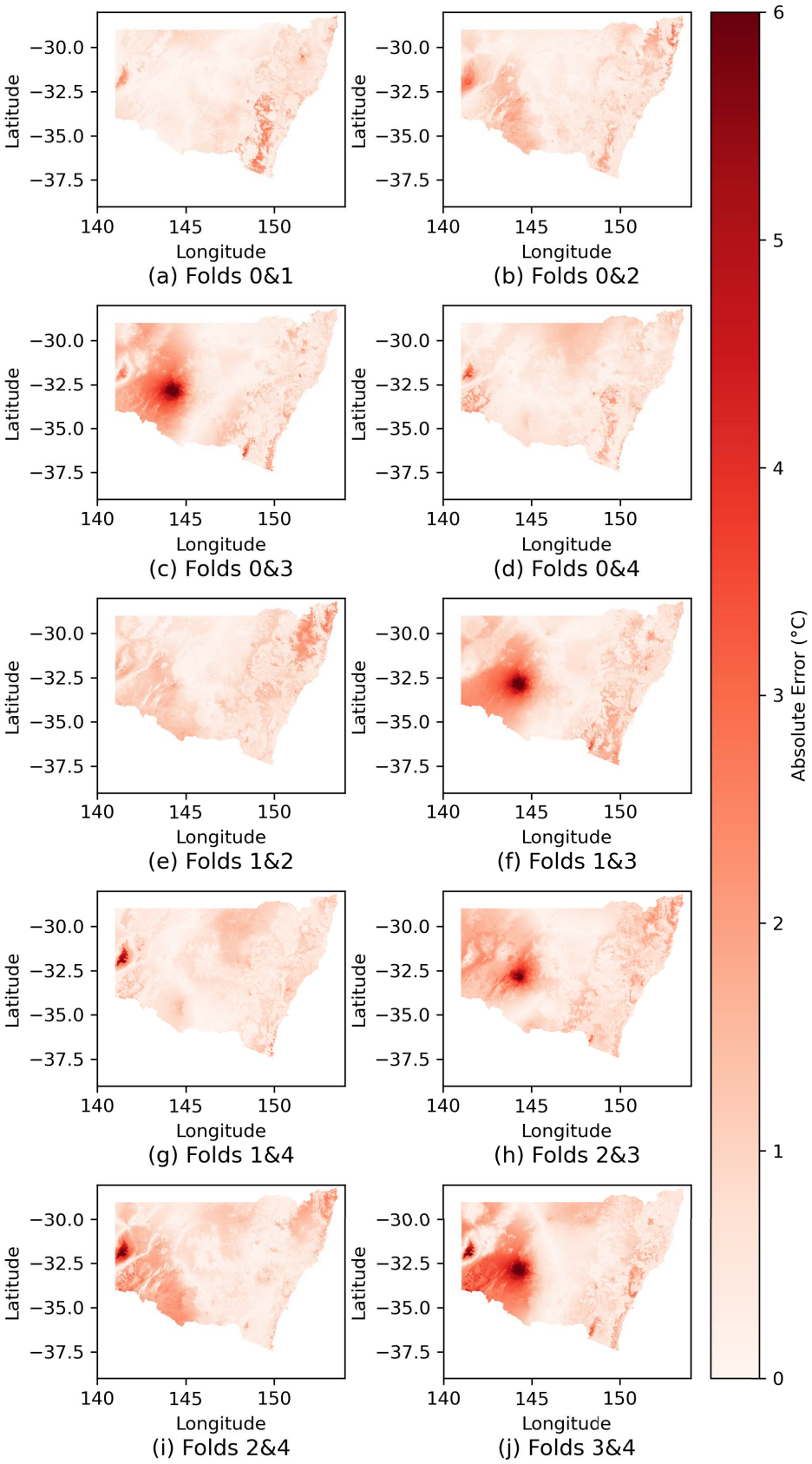


Figure B.5: Absolute Error Map from Weighted Averaged Results Between Folds.