INFORMATION FUSION FOR AUTONOMOUS INTERNET OF THINGS

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Science and Engineering

2021

Tianwei Dai
Department of Electrical and Electronic Engineering
# Contents

List of Tables .......................... 7

List of Figures .................................. 8

Symbols ........................................ 12

Abbreviations .................................. 14

Abstract ........................................ 16

Declaration ..................................... 17

Copyright Statement .......................... 18

Publications .................................... 19

Acknowledgements .............................. 20

1 Introduction .................................. 22

1.1 Background ................................ 22

1.2 Motivation ................................ 24
## 2 Literature Review

### 2.1 Introduction

### 2.2 Development of Information Fusion for Internet of Things

- 2.2.1 Outlier Clearance in Data Preprocessing
- 2.2.2 Event Classification for Wireless Sensor Networks
- 2.2.3 Cooperative Decision-making in Wireless Sensor and Actuator Networks

### 2.3 Preliminaries

- 2.3.1 Consensus
- 2.3.2 Consensus Control Theory
- 2.3.3 Consensus-based Filters
- 2.3.4 Consensus Problem in Wireless Sensor Networks
- 2.3.5 K Nearest Neighbors Rule
- 2.3.6 Deep Learning
- 2.3.7 Deep Reinforcement Learning

### 2.4 Summary

## 3 Online Distributed Distance-based Outlier Clearance Approaches for Wireless Sensor Networks
3.1 Introduction ............................................. 49
3.2 Problem Formulation ................................. 51
3.3 Definitions and Properties of the
    Distance-based Outlier Factor ...................... 52
    3.3.1 Formal Definitions of WADOF .................. 52
    3.3.2 Properties of Weighted Average Distance-based Outlier Factor
        (WADOF) ........................................ 53
3.4 Online Distributed Outlier
    Clearance Approaches ............................... 58
    3.4.1 Top-n WAD .................................... 58
    3.4.2 Adaptive Top-n WAD .......................... 58
3.5 Synthetic and Experimental Results Analysis ..... 61
    3.5.1 Synthetic Sensor Measurements Tests .......... 62
    3.5.2 Advanced Synthetic Sensor Measurements Tests .... 63
    3.5.3 Real Experiments Sensor Measurements Tests I .... 67
    3.5.4 Experimental Setup for Tests I ............... 68
    3.5.5 Results and Analysis for Tests I .............. 71
    3.5.6 Real Experiments Sensor Measurements Tests II .... 72
    3.5.7 Experimental Setup for Tests II ............... 73
    3.5.8 Results and Analysis for Tests II .............. 75
3.6 Discussion ............................................. 77

4 Event Classification in Fixed Topology Wireless Sensor
4.1 Introduction ................................................. 88
4.2 Problem Formulation ....................................... 90
4.3 Distributed 1D-CNNs Algorithm .......................... 91
   4.3.1 1D-CNNs for Feature Extraction ..................... 92
   4.3.2 Distributed 1D-CNNs Process ......................... 94
   4.3.3 Convergence Analysis ............................... 96
4.4 Evaluation Results and Analysis
   on Real-life Datasets ..................................... 100
   4.4.1 Experimental Setup ................................ 100
   4.4.2 Results and Analysis ............................... 102
4.5 Discussion ............................................... 107

5 Coordinated Sensing Coverage in Wireless Sensor and Actuator
Networks with Distributed Deep Reinforcement Learning 109
5.1 Introduction ............................................. 109
5.2 Problem Formulation ..................................... 111
5.3 Distributed Deep Reinforcement Learning for
   Coordinated Sensing Coverage .......................... 112
   5.3.1 Distributed Deep Reinforcement Learning Process . 113
5.4 Evaluation Results and Analysis ....................... 115
   5.4.1 Experimental Setup ................................ 115
   5.4.2 Results and Analysis ............................... 117
5.5 Discussion ................................................................. 118

6 Conclusions and Future Works 121

6.1 Final Conclusions ...................................................... 121

6.2 Future Works .......................................................... 123

Bibliography 125
List of Tables

3.1 Different local outliers detection methods comparison ............... 61
3.2 Sensirion SHT75 .................................................. 73
3.3 Outliers cases for temperature dataset ................................. 80
3.4 Outliers cases for relative humidity dataset ............................ 80
5.1 Different environment settings ......................................... 116
List of Figures

2.1 The very basic CNNs architecture for image classification. . . . . . . . 44

3.1 Sensor node $i$ with its five nearest neighbors. . . . . . . . . . . . . . 52
3.2 Synthetic data without outliers ($T(t)$). . . . . . . . . . . . . . . . 63
3.3 Synthetic data with noise and outliers ($T(t)$). . . . . . . . . . . . . . 63
3.4 Processed results with T-2 WAD ($T(t)$). . . . . . . . . . . . . . . 64
3.5 Processed results with AT-2 WAD ($T(t)$). . . . . . . . . . . . . . . 64
3.6 Synthetic data without outliers ($T_1(t)$). . . . . . . . . . . . . . . 65
3.7 Synthetic data without outliers ($T_2(t)$). . . . . . . . . . . . . . . 65
3.8 Synthetic data with outliers ($T_1(t)$). . . . . . . . . . . . . . . . . 66
3.9 Synthetic data with outliers ($T_2(t)$). . . . . . . . . . . . . . . . . 66
3.10 Measured data in 30.7 sec with outliers for $T_1$. . . . . . . . . . . 67
3.11 Fused data by node $i$ with Top-1 KNN, LOF and WAD for $T_1$. . . . 67
3.12 Fused data by node $i$ with Top-1 KNN, LOF and WAD for $T_2$. . . . 67
3.13 Fused data by node $i$ with Top-2 KNN, LOF and WAD for $T_1$. . . . 68
3.14 Fused data by node $i$ with Top-2 KNN, LOF and WAD for $T_2$. . . . 68
3.15 The smart entity consists of DHT11 and Raspberry Pi. ....... 69
3.16 Target tracking with single Logitech HD Webcam C270. ....... 70
3.17 Object disturbance on webcam. ........................................ 71
3.18 Object disturbance on target. ........................................... 72
3.19 Vertical flash disturbance on tracking. ............................... 73
3.20 Slant flash disturbance on tracking. .................................... 74
3.21 Raw temperature measurements collected by the environment monitoring system. .................................................. 75
3.22 Raw relative humidity measurements collected by the environment monitoring system. .................................................. 75
3.23 Processed temperature measurements with WAD method. ....... 76
3.24 Processed relative humidity measurements with WAD method. .... 76
3.25 Raw relative X data collected by our tracking system. .............. 77
3.26 Raw relative Y data collected by our tracking system. .............. 77
3.27 Processed relative X data with consensus-based filters only. ....... 78
3.28 Processed relative Y data with consensus-based filters only. ....... 78
3.29 Processed relative X data with consensus-based filters and our proposed method. ...................................................... 79
3.30 Processed relative Y data with consensus-based filters and our proposed method. ...................................................... 79
3.31 The deployment of closed neighbourhood on the EPFL campus. .... 80
3.32 Sensirion SHT75 adapted from the Internet. ......................... 81
3.33 Original temperature datasets (Unit °C) ................................. 81
3.34 Full evaluation temperature datasets (Unit °C) ......................... 82
3.35 Original relative humidity datasets ....................................... 82
3.36 Full evaluation relative humidity datasets ............................... 83
3.37 TPR for temperature datasets ............................................. 83
3.38 FPR for temperature datasets ............................................. 84
3.39 TPR for relative humidity datasets ...................................... 84
3.40 FPR for relative humidity datasets ...................................... 85
3.41 ROC curve for temperature datasets ..................................... 85
3.42 ROC curve for relative humidity datasets ............................... 86
3.43 ROC analysis for different typical methods on temperature datasets 86
3.44 ROC analysis for different typical methods on relative humidity datasets 87

4.1 The proposed 1D-CNNs architecture ..................................... 93
4.2 The structure of QUGS ..................................................... 101
4.3 The communication topology of sensor networks ....................... 102
4.4 The PRoD distributions for 12 damage cases when adopting CONS- RVFL framework based 1D-CNNs .............................................. 103
4.5 The PRoD distributions for 12 damage cases when adopting proposed distributed 1D-CNNs .................................................. 104
4.6 The PRoD distributions for 12 damage cases when adopting centralized 1D-CNNs ............................................................. 105
4.7 The training time analysis for different nodes of networks .......... 106
4.8 The memory usage for labeled training data analysis of different nodes of networks. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106

5.1 The fixed undirected communication graph. . . . . . . . . . . . . . 115

5.2 The deployment and final actions of all entities for Setting 1. . . . . . . 117

5.3 The deployment and final actions of all entities for Setting 2. . . . . . . 118

5.4 Loss for Setting 1. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119

5.5 Loss for Setting 2. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
Symbols

$E(\cdot)$  The Mathematical expectation function

$\mathbb{R}$  The real field

$\mathbb{Z}$  The integers

$\mathcal{A}$  The adjacency matrix of multi-agent systems

$\mathcal{D}$  The degree matrix of multi-agent systems

$\mathcal{E}$  The edges of multi-agent systems

$\mathcal{G}$  The communication graph of multi-agent systems

$\mathcal{L}$  The Laplacian matrix of graph

$\mathcal{V}$  The nodes of multi-agent systems

$\otimes$  Kronecker product

$\rho(\cdot)$  The spectral radius

$1D\text{conv}(\cdot)$  The convolutional operation

$\text{MaxP}(\cdot)$  The max-pooling operation

$P(\cdot)$  The Probability function

$A_{k_i}^2$  The 2-permutation of $k_i$

$C_{k_i}^2$  The 2-combination of $k_i$

$F^*$  The common global learning model
$f_i^*$  The local optimal learning model

$\text{Edist}(\cdot)$  The Euclidean distance
Abbreviations

AI  Artificial Intelligence.

AIoT  Autonomous Internet of Things.

ANNs  Artificial Neural Networks.

BoAT  Blockchain of Autonomous Things.

CNNs  Convolutional Neural Networks.

FPR  False Positive Rate.

IoT  Internet of Things.

KNN  k Nearest Neighbors.

LOF  Local Outlier Factor.

LWADOF  Lower Bound of the Weighted Average Distance-based Outlier Factor.

PRoD  Probability of Damage.

ReLU  Rectified Linear Units.

RL  Reinforcement Learning.

ROC  Receiver Operator Characteristic.

SGD  Stochastic Gradient Descent.

SVM  Support Vector Machine.
**TPR** True Positive Rate.

**WAD** Weighted Average Distance.

**WADOF** Weighted Average Distance-based Outlier Factor.

**WSANs** Wireless Sensor and Actuator Networks.

**WSNs** Wireless Sensor Networks.
The Internet of Things (IoT) extends the electronic connectivity into millions of IoT nodes around the world. In order to achieve the autonomy to make sound decisions based on the collected and analysed information, artificial intelligence algorithms are for networks to leverage to turn into autonomous IoT (AIoT). The core value of these AIoT applications is from analysing the information provided by these smart entities and performing autonomous decision-making. However, general issues in networks, such as the outlier raging, communication overhead, memory shortage, etc., render the information fusion process ineffective, which leads to the unreliable AIoT applications. Thus, the advanced information fusion methods must be proposed to address such issues for the sake of fully utilizing these smart applications and services.

To solve such real-life problems, various information fusion approaches are developed for the AIoT systems. For the outlier raging problems in AIoT applications, the distance-based method inspired by the KNN rule has been used. In addition, artificial intelligence algorithms, such as the deep learning and reinforcement learning, are integrated with consensus theories to handle the distributed learning problems in AIoT applications for the sake of solving the communication overhead and memory shortage in networks. Also, the comprehensive evaluation works show the proposed approaches emerge the powerful capability, and they provide important operational advantages over traditional centralized methods and typical distributed approaches.

The main contributions of this thesis are proposing two outlier detection methods with compelling features, such as low computational complexity and memory usage, for the outlier raging in networks, and one novel distributed learning framework that can save the average training time and memory storage compared with the typical methods.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s Policy on Presentation of Theses.
Publications

Journal Publications:

1). Tianwei Dai and Zhengtao Ding. “Online Distributed Distance-based Outlier Clearance Approaches for Wireless Sensor Networks.” Pervasive and Mobile Computing (63) 2020


Conference Publications:


First and foremost, I would like to thank Professor Zhengtao Ding for his supervision in these years. He led me through the door of information fusion and pointed out the new research field to form my PhD project.

I would also like to say thank you to all control systems centre colleagues who helped me to solve the difficulties that I encountered during my studies.

Last but not least, thanks must go to my parents. I probably would have not completed this project without their support.
To Dai, Hongbin and Han, Ling
Chapter 1

Introduction

1.1 Background

There is a popular supposition that we have been dragged into a new age of digital information. More specifically, a world with features that nearly all things have worked clandestinely with their locations and functional states unknown or even unknowable has changed to a world with features that a large number of homogeneous or heterogeneous internet-connected entities enable near-ubiquitous monitoring of the states of interest, such as buildings, environments, people, etc. The internet of things (IoT) is making an extraordinarily profound impact on our lives. Since almost everything becomes connected, it is changing the way society functions and bringing the great economic benefit, which is worth trillions of dollars and increasing year on year [1–3].

Nowadays, as the global accelerating urbanisation, a city faces a series of critical issues including population explosion, deteriorating environment, energy exhaustion, traffic congestion, etc., thus, more and more attention is paid to the future city concept. There is not a clear and consistent definition for the future city as yet, but many do agree that it will be smart. In many ways, the smart city concept can be simply regarded as good urban development and planning that seeks to integrate the advanced information and communication technology with the new thinking in public transport
facilities, building security monitoring, entertainment tourism, environmental sustain-
ability, good governance, and transparency. Generally, urbanites in the intelligent city
are able to run traditional networks and services more flexibly, efficiently and sus-
tainably through a new layer of electronic connectivity formed of the IoT [4]. In the
last decades, as the cost of bandwidth, computing, memory, sensors, etc. has fallen
dramatically around the world, we have witnessed the formation of the early stage
of the IoT, which was getting devices connected and making them smart. However,
in the wake of developments in both analytic methodology and Artificial Intelligence
(AI), we are right on the edge of the next wave of IoT, which is about achieving the
autonomy for those smart devices to be capable of sensing, understanding, adapting
and reacting to the physical world. In other words, IoT devices turn into Autonomous
Internet of Things (AIoT) by integrating those devices and machine learning algo-
rithms. The autonomous IoT (AIoT) applications and related services in the real
world are beginning to take shape, such as vehicular computing and networks in in-
telligent transportation [5–9], energy trading and management in smart grid [10–13],
communication and energy consumption optimisation in Wireless Sensor and Actuator
Networks (WSANs) [14–16].

Generally, the AIoT networks have the dynamic perception of the external environment
and can make related control decisions to react. Wireless sensor and actuator networks
(WSANs), in essence, are the perfect implementation environment of the AIoT con-
cept, where the AIoT devices with sensors and actuators are deployed. WSANs are a
large number of low-cost wireless devices with the ability to real-time sense the physi-
cal world (events) and rapidly execute relatively complicated actions (solutions), based
on the collected, shared and fused data by smart entities. More specifically, after de-
ployed, sensor nodes are in charge of accumulating the information of interest, carrying
out local processing of these data including information cleaning, compression, broad-
casting, fusion, analysis, etc., and generating control decisions for actuators to react.
Actuators, on the other hand, may have a simple mechanical structure while possessing
sufficient computing power and energy resources so as to perform complex tasks
such as valve operation, turntable control, gearbox shift, etc. Recently, WSANs have
found an increasingly wide utilisation in different fields such as pollution management,
building automation, smart grid, intelligent transportation, etc. [17–20]. In fact, information fusion across entities in networks can improve the information completeness and estimation quality of states of interest, enhance the efficiency of actual operation, and drive incremental advances in optimizing resource management of networks.

In fact, the main task of the AIoT is collecting the states of interest. However, entities in networks may often provide the imperfect estimates or just outliers due to their placements in very harsh environments or due to the inherent challenges in measuring and estimating the variable of interest [21]. Aggregating estimates across entities in networks is one of the most powerful approaches to promoting the estimation quality. More specifically, information fusion is concerned with how best to efficiently transfer and aggregate measurements across a pre-determined set of entities that share information. Also, the information fusion is widely treated as a remedy for improving the accuracy of event classification and the efficiency of decision-making in an extensive variety of AIoT applications, such as power systems, traffic control, smart building, and various other industrial areas [22–25]. Additionally, the centralized methods predominate in the context of traditional information fusion approaches due to the simple execution structure and reduced internal conflict. Nevertheless, in recent years, the distributed information fusion approaches with many advantages have exhibited bright prospects in improving the efficiency of actual operation and optimizing resource management for the large networks compared with the centralized methods [26]. In this thesis, we concentrate on these motives for information fusion across the entities in networks. Different distributed information fusion methods are proposed to solve the outlier clearance problems and event classification and decision-making efficiency problems in the AIoT applications.

1.2 Motivation

In fact, for the sake of the superior design and implementation of WSANs, we must handle many critical issues arising in WSANs such as the outlier raging, event classification, sensing coverage problem, etc. The problems mentioned above are three of the most principal problems addressed in WSANs literature [27–35].
Owing to the imperfect nature of WSANs, the quality of raw measurements collected by smart entities is usually inaccurate and unreliable in practical applications [27], which further adversely affect information fusion results and decision-making processes. It is worth noting that detecting and clearing those spurious and imprecise raw measurements in WSANs become two of the first and foremost prerequisites for successfully ensuring the high-accuracy and high-reliability in monitoring system and being able to make effective and sound decisions with information fusion. Obviously, the high-quality raw measurements make a positive effect on the event classification process in the AIoT applications. Hence, plenty of research works [36–40] have been done to solve this problem in recent years. [36] and [37] are the typical centralized approaches, which have the simple structure and are easy to implement for the small system. However, if its base station crashes, the entire system may go down and the capability of it will be the performance bottleneck in a large system. And the distributed approaches [38–40] are proposed to improve WSANs, which save the computational power, communication cost, memory storage, etc. But there are some drawbacks for the current typical distributed approaches. For instance, [39] with Top-n style needs the prior knowledge about the outlier distribution. Also, the SVM method [40] demands a great deal of computing and memory to preserve its high accuracy of detection. Advanced approaches without such limitations should be proposed.

In addition, the event classification approach allows the efficient management of emergencies and critical incidents owing to the occurrence of certain events, which can lead to the substantial improvement of WSANs in the smart city. For the sake of the improvement of event detection rate, fault tolerant, etc., different event classification approaches are proposed. With reference to the centralized methods, they have the same downsides which are mentioned above. Hence, the distributed methods are the main stream in this scope. Recently, more and more researchers are working on the development of AI-based event classification approaches for WSANs. One of the most important advantages of AI-based approaches is that they are able to configure and adjust their sets of parameters independently instead of requiring specialized knowledge. And some distributed learning methods, such as [41–43], are proposed so as to
reduce the event detection delay, training time, computing capability and memory usage. However, all the above distributed learning algorithms just set the learning phase apart from the consensus phase for the simplification. Such framework may suffer from the asynchronous termination. Furthermore, it would require more time to finish the learning when more consensus steps are executed overall. Since, it is finished by two continuous processes that are independent each other where the consensus process is slow compared with the local learning. The new advanced framework, which combines the learning phase with the consensus phase, should be proposed.

There are many critical issues in WSANs, such as the network delay issue, the sensing coverage problem, the energy consumption optimisation, etc. The sensing coverage problem is one of the most important problems in WSANs. In the sensing coverage problem, the main task for networks is to appropriately cover the target environment so as to guarantee that all pre-defined events which occur in that area can be reliably, accurately and efficiently identified by networks with the minimum energy consumption. One robust and powerful method without requiring specialized knowledge should be proposed.

Motivated by the above problems, different new approaches are presented in this thesis which emerge powerful capability and provide important operational advantages over traditional centralized and distributed approaches. These associated control algorithms constitute the purpose of this thesis.

### 1.3 Contributions

The contributions of this thesis are listed as follows:

- There are shortcomings for traditional distributed outlier clearance approaches which are introduced in Chapter 2. Different from traditional approaches, novel distance-based concepts are presented to design the outlier factor in networks.

- Based on this novel outlier factor, two different online outlier clearance approaches with low computational complexity and memory usage are established.
and tested on the real-life datasets with other typical approaches. Moreover, the analysis of the properties of the new outlier factor is developed.

- There are some drawbacks to traditional learning frameworks which are introduced in Chapter 2. Compared with traditional frameworks, a novel distributed learning framework is presented which emerges powerful capability and provides important operational advantages. Besides, the convergence analysis of this distributed learning process is developed.

- Integrating our novel distributed learning framework with the deep learning method, an approach with compelling features to solve the distributed deep learning problems in event classification for WSANs is presented. The proposed approach and other typical methods are tested on the real-life structural health monitoring benchmark datasets. And the evaluation results show that our proposed approach achieves competitive performance with high accuracy and reliability, and low learning time and memory usage.

- Combing our novel distributed learning framework and deep Reinforcement Learning (RL) method, a new distributed decision-making approach for the coordinated sensing coverage problems in WSANs is presented. Simulation works indicate that it achieves competitive performance with compelling features.

1.4 Thesis Outline

The organisation of each chapter is described in detail, which is at the very beginning of that chapter. To understand the whole thesis structure, a general overview is presented as follows:

Chapter 1: Brief overviews of the AIoT applications, and the influences and developments of them are shown in this chapter. Meanwhile, the motivation and contributions are also given.

Chapter 2: The existing results are reviewed in this chapter, including outlier clearance methods, AI-based event classification methods and decision-making methods for the
AIoT applications. Furthermore, some related preliminaries are introduced, such as the consensus control theory, K nearest neighbour rule, deep learning methods, deep reinforcement learning algorithms, etc.

Chapter 3: One key challenge for the AIoT applications is to provide real-time high reliable measurements with the minimum resource consumption. Outlier clearance in networks can ensure the quality of measurements and dependable monitoring. In this chapter, we propose two online distributed outlier clearance approaches with low computational complexity and memory usage that can identify and remove the spurious measurements. The proposed approaches are operated locally and thus save communication overhead as well as possess good scalability. The evaluation performance of proposed approaches and existing widely used methods on the synthetic and real-life datasets illustrate that our Adaptive Top-n Weighted Average Distance (WAD) approach achieves remarkable outlier clearance performance as compared to these methods.

Chapter 4: In this chapter, we propose an approach with compelling features to solve the distributed deep learning problems in event classification for the AIoT applications. Each entity is able to obtain its local learning model with its private annotated data. Our aim is to make every node converge to a common global learning model via the communication only among its neighbours, where neighbour relationships are described by a fixed and connected graph. The proposed approach is tested on the structural health monitoring benchmark datasets. Evaluation results show that our proposed approach achieves competitive performance in test datasets with high accuracy and reliability compared with other typical methods. Besides, it eliminates the need for transmitting and synchronizing measurements at central entities and also reduces the average training time and the required memory usage for each entity compared with centralized methods.

Chapter 5: IoT extends the electronic connectivity into millions of IoT nodes in our city, which collect, share and fuse information to comprehend the status of the city. In order to achieve the autonomy to make control decisions based on the collected and analysed information, a promising artificial intelligence method, RL, is for smart
entities to leverage. In this chapter, we propose a distributed learning approach using
the deep RL method and consensus theories to solve the coordinating sensing coverage
problems in WSANs. Also, evaluation works show the proposed algorithm emerges
powerful capability.

Chapter 6: This chapter includes the contributions and the main results of this thesis,
and the possible future works are also presented in the conclusion.
Chapter 2

Literature Review

2.1 Introduction

IoT is emerging as a vital platform for innovation where connected entities generate data, then, enable smart services [44–47]. These entities disseminate information which is obtained and utilized by smart services. The real value of these IoT smart services comes from analysing the data provided by these entities. The completeness of the information and the quality of the service can be improved by fusing information across multiple entities. In this chapter, the development of the related methods and some significant results are reviewed. Moreover, corresponding preliminaries are also presented.

2.2 Development of Information Fusion for Internet of Things

2.2.1 Outlier Clearance in Data Preprocessing

The IoT systems usually consist of a large number of smart entities, each of which is low power device usually equipped with one or more sensors and other modules. In most
IoT applications, it is required to monitor targets and analyse sensor fusion data in real-time. However, the raw measurements collected by smart entities may be inaccurate and unreliable in such practical applications [48]. On one hand, the capability and resources of entities are stringently limited, such as energy supply, internal memory, etc. On the other hand, in practical applications, numbers of entities in networks are often randomly or methodically deployed in enormous and harsh natural environments [49]. In addition, entities may face malicious attacks from the enemy, such as black hole attacks, sensor nodes capture and communication denial of service attacks [50]. Consequently, the aforementioned internal and external complications make a sharp decline in the quality of raw sensor measurements, which further adversely impact sensor fusion results and decision-making process [51, 52]. It is obvious that one of the main prerequisites for successfully ensuring the high-accuracy and high-reliability of sensor measurements and being able to make effective and sensible decisions with sensor fusion information is online identifying and clearing erroneous data in networks with minimum energy consumption.

Outlier detection and clearance approaches in Wireless Sensor Networks (WSNs) aim to ameliorate the quality of raw sensor measurements in order to deliver the optimal information to the decision-maker. Due to outlier detection and clearance approaches have great potential for the application in sensor networks, recently, many of them peculiarly developed for sensor networks have emerged. Outlier detection and clearance encompasses aspects of a broad spectrum of techniques, which are fundamentally identical but with dissimilar names, for instance, anomaly detection, deviation detection, exception mining or fault diagnosis. According to [27, 53, 54], the existing typical approaches can be sorted as centralized approaches, decentralized approaches and hybrid approaches based on the architectural structure. Moreover, on account of the disciplines, a broad range of outlier detection and clearance methodologies can be categorized into statistical-based, nearest neighbour-based, clustering-based, classification-based, and spectral decomposition-based approaches [55]. Furthermore, statistical-based methods are essentially model-based techniques, in which they estimate the probability distribution model of sensor measurements to identify the error
sensor measurements. Additionally, they can be further categorized into parametric and non-parametric methods based on how the probability distribution model is estimated. Bayesian network-based and support vector machine-based methods belong to classification-based domains. They learn a classification model based on the set of training instances and classify an arriving sensor measurement into one of the learned class, normal class or outlier class. Spectral decomposition-based approaches use principle component analysis for outlier detection and clearance.

The central type method is one of the most commonly used methodologies in outlier detection and clearance within the small-scale WSNs. It identifies the outliers through the entire sensor measurements stored at a base station. Following this architecture, Curiac et al. build an ensemble system with multiple classifiers, which are selected to implement a complex decisional system in the base station, to clear the sensing anomaly in WSNs [36]. In addition, a central type method, presented in [37], utilizes complex transforms of multivariate time-series to produce accurate models and further enhance its performance in fault detection and clearance.

However, traditional central type methods, such as [36] and [37], cannot deal with the scalability issues, high communication cost, and huge memory overhead in the exceptionally large-scale WSNs. Because of the increasing sensor nodes deployed in WSNs, it requires a large amount of storage space and costly processors with great computational capacity as the fusion centre. Nevertheless, distributed approaches generally demand less computational power, lower communication cost, and smaller memory footprint [56–58]. In [38], Branch et al. propose a global and semi-global nearest neighbour-based outlier detection and clearance approaches for the sake of identifying Top-n outliers. Despite requiring the prior knowledge about outliers to define Top-n value, it demands less energy supply and bandwidth compared with the well-known centralized outlier detection algorithm, AODV [59]. Concerning the local outlier in WSNs, some nearest neighbour-based approaches and classification-based techniques can be employed to identify them. A prime example of nearest neighbour-based approaches is that of the local outlier factor [39]. Local Outlier Factor (LOF)-based approaches with the hierarchical network structure, proposed in [60], are employed to identify anomalous sensor measurements. Also, a simplified variation of LOF applied
in WSNs is described in [61]. Abid et al. create one distinctive nearest neighbour-based outlier detector, DNOD [62]. This unsupervised detector can identify outliers without defining the number of nearest neighbours and the distance threshold of accepted neighbours. Nonetheless, its high accuracy of detection is guaranteed by the large size of the learning window, which significantly increases storage and computing. Zhang et al. propose different distributed outlier detection approaches in [40] and [63] using the Support Vector Machine (SVM). They can make each sensor node sequentially update the normal behaviour model of itself using meaningful information derived from the spatial and temporal correlations between itself and each of its spatially nearest neighbours in order to identify local outliers. Furthermore, an innovative semi-supervised SVM classifier described in [64] can identify the local outlier of solar power in the large photovoltaic power stations. Although these SVM-based approaches prove a high detection accuracy and low false alarm rate, they have the issues in requiring the high computational capacity and memory storage, which restricts the application of real-time scenario.

2.2.2 Event Classification for Wireless Sensor Networks

Event-driven strategies for WSNs are motivated by the large-scale application of inexpensive sensor nodes with limited resources in related areas. As one of the advantages of event-driven sensing technologies is saving great energy, they are widely applied in WSNs, where the network centres transmit the collected information only when the pre-defined event occurs [65–68]. Generally, specific events will be defined in advance according to user requirements, referred to as the pre-defined events, denoting the status of interests. Subsequently, for the successful implementation of event-driven strategies, the high accuracy and reliability of event classification are becoming increasingly important.

Recently, with steady advances in varied sensor technologies, the studies and applications of information fusion have been continuously extended forward to a further and wider range as multidisciplinary areas. Meanwhile, it has witnessed a increasing number of information fusion-based approaches and methods for the event classification in
WSNs proposed to improve the performance of networks. A multiplemetric learning algorithm, which can be used to learn a set of optimal homogeneous (or heterogeneous) metrics, is proposed by [69] to fuse the information and classify the events. Fortino et al. [70] presents a framework for collaborative computing and multisensor data fusion in body multisensor networks monitoring a single individuals’ emotion. Additionally, for the purpose of protecting forest resources, Zervas et al. [71] constructs a fire detection system utilizing heterogeneous multisensor data.

Actually, event classification approaches for WSNs must possess the high accuracy, strong reliability, fault tolerant, real-time characteristic, and low resources consumption [72, 73]. In the context of conservative approaches, such as [30, 31, 66–68], most of the sensor nodes in WSNs are confined to measuring and transmitting the raw data to base stations or gateways, which are referred to as the resource-rich central entities having sufficient competence in further data analysis and event identification. Nevertheless, the natural characteristics of WSNs, such as multi-hop communications, switching topologies, and low bandwidth availability, result in many remarkable challenges when designing the centralized techniques for WSNs. Challenges include, but are not limited to: events detection delay, packets loss, networks congestion, etc., overwhelmingly increasing the difficulty of centralized techniques applications in the smart city.

Besides, reporting all measurements from the whole networks to the base stations is expensive and unreliable. Consequently, users want to perform the event classification inside WSNs in order that the raw data can be analysed by the sensor nodes locally. For example, some distributed approaches introduced in [72] satisfy such requirements. However, in these approaches, usually, a group of sensor nodes is involved in the data processing that requires the data exchange amongst sensor nodes. Naturally, it incurs the high communication overhead when performing the event classification through cooperative distributed data processing. Hence, it expects that the single sensor node is capable of executing the data processing task individually in order to slash the huge communication cost. In other words, it considerably reduces the event detection delay. Then, actuators residing in WSNs are able to react to detected events much faster, accurately and reliably. Also, it expects that the demand for computing capability
and memory usage for each sensor node in networks can be as low as possible, which trims the whole cost of implementation of WSNs.

In recent years, a great deal of research work has been done on the AI-based event classification approach for WSNs [74, 75]. One of the most important advantages of AI-based approaches is that they are able to configure and adjust their sets of parameters independently instead of requiring specialized knowledge [76]. Respective AI algorithms can be further classified into supervised and unsupervised categories. More specifically, the supervised algorithm can finish the iterations of parameter configuration with information extracted from the annotated datasets during the training phase. With respect to the unsupervised algorithm, it can draw inferences from datasets consisting of inputs without labelled responses. In addition, the implementations of AI-based techniques in WSNs are inclined to be less computationally intensive compared with others, for instance, probabilistic/statistical model-based approaches [77]. It is well known that deep learning is one of the foremost parts of a broader family of AI algorithms based on the learning data representations and successfully applied to fields including computer vision, speech recognition, machine translation and gaming [78]. Yet, as often as not, we notice that the related theories have rarely been used for solutions to the event classification problems in WSNs. In [30], Chao Tong et al. exhibit the similarity in the form between the image and the sensor measurement, then propose a Convolutional Neural Networks (CNNs)-based method to enhance the event classification accuracy for sensor networks. Also, a novel WSNs-based water leakage detection system designed with the one dimensional (1D)-CNNs and the SVM is introduced in [31]. Nevertheless, they are the centralized solutions that the resource-rich central entities are required and the communication cost would be extremely expensive. In recent times, Avci et al. develop a pioneering WSNs-based structural damage detection system in [79] which train the individual 1D-CNNs for each sensor node in networks to perform the event classification locally. Obviously, it is able to eliminate the need for data transmission and synchronisation in the event classification phase, while this centralized 1D-CNNs leads to the long training time and high memory usage in the training phase. In fact, nowadays, it is quite common for users to collect, store
and process data locally as data exchange is not possible due to the privacy considerations. In such a circumstance, Georgopoulos et al. propose an algorithm to learn from the distributed data on a network with an arbitrarily connected graph without the exchange of the training data [41]. And, in [42], two distributed learning algorithms for training random vector functional-link (RVFL) networks, which can be viewed as feedforward neural networks with a single hidden layer, through interconnected nodes are proposed where training data were distributed under a decentralized information structure. However, all the above distributed learning algorithms may suffer from the asynchronous termination that the stop criteria at some agents may be valid for termination but other agents have not reached the same decision. In addition, it can be expected that once they execute more consensus steps overall the convergence rate of them will be quite slow as they comprise two successive processes where the consensus process is slow compared with the local learning.

2.2.3 Cooperative Decision-making in Wireless Sensor and Actuator Networks

Within the context of traditional methods, all the devices in WSANs are obliged to transmit their raw data to the base stations or gateways, which possess strong competence in data analysis and decision-making [32,34,80]. Then, the base station produces control decisions for the actuators to implement. However, the particular features of WSANs, multi-hop communications, switching topology, low bandwidth availability, etc., lead to plenty of remarkable challenges when designing the centralized method for WSANs such as execution delay, unforeseeable link damaged, networks congestion, etc. Beyond, sending out all information from deployed entities to the base station is extremely expensive and unreliable. Accordingly, a distributed method to the sensing coverage problem is much more attractive compared to a centralized one. On the one hand, sensing and executing entities are normally decentralized spatially across the target environment, thus a distributed approach is more natural to be implemented. On the other hand, owing to the inherent attributes of the distributed approach, it provides great convenience in the implementation of WSANs such as high scalability and
availability, less memory, computational capabilities and communication bandwidth demand, etc. Consequently, so far, plenty of distributed approaches are proposed to achieve a reliable, efficient, resource-aware and scalable solution to the sensing coverage problem in WSANs [81–85].

In addition, for the sake of the autonomous sensing and executing, reinforcement learning (RL) algorithms lead ambient intelligence into WSANs via offering a class of solutions to the closed-loop problems in the operation of WSANs from information analysis to decision-making to action performed. More specifically, RL algorithms would enable sensing and executing entities to learn the optimal policy that maps states to actions through the interaction with the target environment [86–88]. In other words, the trained entity is able to observe the full or partial status of the target environment via sensors and perform related actions to deal with the current situation so as to maximize the long-term reward. In spite of the successful application of RL approaches in related domains, they still confront principal challenges when dealing with real-life complicated tasks. For example, the learning entities in the real world have to efficiently and precisely depict the status of the target environment with high-dimensional sensor information in order to learn the optimal policy. In recent times, Mnih et al. propose an approach, the deep RL, integrating RL with deep learning to overcome the curse of dimensionality [89]. It becomes widely known throughout the world after they consecutively defeated the best human professional Go players with the AlphaGo, which is based on the deep RL.

2.3 Preliminaries

2.3.1 Consensus

Networked dynamic agents system has been the main object of study in distributed artificial intelligence since the 1980s [90]. The ultimate aim of researches in the coordination of multi-agent systems is to complete the sophisticated mission utilizing the multi-agents, which are relatively simple and inexpensive. Due to the low-cost,
reliability, robustness, high efficiency and scalability of the multi-agents system, it can be widely applied in a vast number of areas, such as computer networks, intelligent manufacturing, power systems, traffic management and military reconnaissance.

The study of the consensus problem is the key fundamental for a multi-agent cooperative control scheme. As a result, formal and systemically researches on consensus problem are indeed worthy. In a multi-agent system, the "consensus" means that certain states of agents in this system would approach consistently over time. A consensus protocol is the primary rule for information exchange in the network, which depicts the process that each agent communicates with its neighbours.

In practical applications, the multi-agent system needs to agree upon certain quantities of interest in order to subdue the internal uncertainties and external environment mutations. Consequently, the agreement of these agents is a prerequisite to achieve the cooperation and it is also important to settle the agreement problem in a multi-agent system with the possibility of communication failure and delay [91].

In [92], [90], and [93], they build the general analysis framework for posing and addressing consensus problems in multi-agent dynamic systems. Further extensions based on Fax’s work are presented by [94] and [91]. Thereafter, in the control system society, it has witnessed great advances in various effective consensus algorithms reaching the agreement. In [95], they propose an approach for solving agreement problems in the study of alignment without computing any objective functions. Compared with the routine consensus protocols, the method from [96] can considerably reduce the time of convergence via predictive mechanisms without switching the topology of networks. Lin et al. [97] focus on the consensus problem for directed networks of agents with external disturbances and model uncertainties using the $H_\infty$ control strategy. A new framework for addressing the consensus of multi-agent systems and the synchronisation of complex networks is introduced by [98], and also a distinct concept, consensus region, is proposed. Khoo et al. [99] present a practical robust finite-time consensus tracking algorithm for a multi-agent system, which is based on the terminal sliding mode control. In recent years, the application of a distributed adaptive control scheme
based on the backstepping technique in multiple nonholonomic dynamic systems attracts great attentions.

In fact, there have been tremendous problems related to multi-agent systems with close ties to the consensus problem. Moreover, on the ground that a great deal of multi-disciplinary research work on the consensus problem has been done in recent years, the consensus algorithm is considerably widely applied in many fields.

2.3.2 Consensus Control Theory

In a WSANs with the communication graph, $G = (\mathcal{V}, \mathcal{E})$, it makes assumptions that every entity $i$ in networks has an associated scalar value $q_i \in \mathbb{R}$, where $i = 1, 2, ..., n$. The arithmetic mean of these values at each entity can be obtained via the consensus algorithm in cooperation with the local connected entities.

The basic discrete-time linear consensus algorithm is

$$q_i(t + 1) = w_{ii}q_i(t) + \sum_{j \in \mathcal{N}_i} w_{ij}q_j(t), \quad (2.1)$$

where $t \in \mathbb{Z}^+ \cup \{0\}$, $q_i(t)$ and $q_i(t + 1)$ denote the time iteration index, the current state and the next state, respectively, and $\mathcal{N}_i$ shows the set of neighbours of entity $i$. In addition, $w_{ij} \in \mathbb{R}^+$ are the coefficients associated with the edges of the graph, which ensure the coherence of the local estimates. Particularly, $w_{ij} \neq 0$ indicates that entities $i$ and $j$ are linked and $w_{ij} = 0$ otherwise.

The coefficients of compliance with the related requirements guarantee the convergence of the algorithm. Furthermore, $W^T = W$, $\mathbf{1}^T W = \mathbf{1}^T$, $W \mathbf{1} = \mathbf{1}$ and $\rho(W-\mathbf{1}\mathbf{1}^T/n) < 1$ are the sufficient conditions for its convergence, where $\mathbf{1} = (1, 1, ..., 1)^T \in \mathbb{R}^n$, the elements of $W \in \mathbb{R}^{n \times n}$ are the coefficients, $w_{ij}$, and $\rho(\cdot)$ is the spectral radius. Then, the state of networks will be asymptotically stable at the arithmetic mean $(1/n)\mathbf{1}\mathbf{1}^T q(0)$, where $q(0)$ is the initial values for all entities, within appropriate iterations $p \in \mathbb{Z}^+$. Actually, the number of iterations, $p$, determines the precision of local estimations and the level of agreement [100].
Up to this point, the consensus concept is applied in a wide range of fields, such as information fusion, drones formation, etc., due to its superior advantages, such as inherently distributed, running without routing tables, packet switching, etc. [101,102].

2.3.3 Consensus-based Filters

The general idea for sensor networks is to observe the underlying process with a group of sensor nodes, which consist of sensing, data analysis and communicating components, configured according to a given network topology. We expect that the measurements from each individual sensor node and its neighbours can be used to make our estimation of states of interest more accurate and reliable.

As a consequence, one of main tasks is to develop distributed sensor fusion approaches with the purpose of estimating the states of interest more reliably and effectively. Consensus-based filters can play the key role in this task with many advantages, such as scalability and working without a fusion centre.

The most basic consensus-based filter [103] in discrete-time is:

\[
q_i(t + 1) = q_i(t) + \delta \sum_{j \in \mathcal{N}_i} w_{ij}(q_j(t) - q_i(t)) + \sum_{j \in \mathcal{J}_i} w_{ij}(u_j(t) - q_i(t))
\]

where \(u_j\) expresses the measurement of sensor node \(j\) and \(\delta\) represents the step-size of iterations. Also, \(\mathcal{J}_i\) is the set of inclusive closed neighbours, respectively.

This consensus approach, consensus on estimates (CE), can reach a collective agreement over networks without requiring the information of error covariance matrix or probability density functions. These are the main reasons that an increasing number of people design an extensive type of consensus filters involving the CE.

The first two-stage CE strategy, involving Kalman filtering, is proposed by [104]. It solves the distributed estimation problem by reducing it to two separate dynamic consensus problems in terms of weighted measurements and inverse-covariance matrices. First, compute the mean of sensor measurements with consensus approaches; second, update and predict the local estimates using the centralized Kalman optimal gains.
But this approach is only applicable to sensors with identical observation matrices. A modified algorithm [105], based on the work from [104], uses two identical consensus filters to address the limitation problems, which enables sensor networks to act as a collective observer for the processes occurring in a case.

Carli et al. [106] consider the problem of estimating the state of a dynamical system from the distributed noisy measurements with the consensus perspective and to construct the estimation. And it is performed over a two-level scheme, the first being a Kalman-like update and the second being an estimate fusion using a consensus matrix. A robust state estimation algorithm for complex large scale systems is presented by [107], which can offer a reliable estimation regardless of the intermittent observations and communication faults. Further, Yu et al. [108] put forward a new case, which is measuring the target information through only a small fraction of nodes, for sensor networks. In this case, a new type of consensus filter was designed with pinning observers. Meanwhile, Matei et al. [109] devise a consensus filter with a Luenberger-type observer in two steps, which is similar to the work from [106]. Lastly, Zhu et al. [110] design a filter based on the CE for the target tracking over the heterogeneous sensor networks.

In addition, the Kalman filter is a popular tool for estimating the state of a dynamic system, given the low quality of sensor measurements. Kalman filter is able to estimate what the corresponding output information is. However, the observed information including outliers would degrade the performance of the Kalman filter. The consensus-based filters have the same issue. Therefore, we need to propose new approaches for outlier detection and removal.

2.3.4 Consensus Problem in Wireless Sensor Networks

Sensor networks technology is one of the most important technologies for the future. Low-cost, reliable and smart sensor nodes with data processing and communicating capabilities, networked via wired or wireless links and spatially distributed in large numbers, rise unprecedented opportunities in military, industry, security and environment areas according to [111] and [112]. Furthermore, sensor networks offer the
technology to an extensive spectrum of systems in civil applications, creating new capabilities for the smart city, intelligent power systems, etc.

Some specific practical applications of sensor networks include: reconnaissance and surveillance, environment monitoring, pipeline leak inspection and detection, air traffic control, manufacturing automation, simultaneous localisation and mapping, multi-target tracking, smart home and health telemonitoring [113–119]. The deployment of sensor nodes in these applications may be underground, underwater or even changing with time, the communication among them may be through wires or wireless links, and the sensor nodes may be small or large. It is obvious that sensor networks for various applications are quite distinct, however, they would deal with many common technical issues.

The general idea for sensor networks is to observe the underlying process with a group of sensor nodes, which consist of sensing, data analysis and communicating components, configured according to the given network topology. And we obtain the estimation of states for the dynamic system based on not only the measurement from each individual observer but also its neighbours’. Generally, in networks, each node with limited data processing capability can only acquire and analyse local information of the states of interest. Moreover, the environment, where the nodes are deployed, is extremely harsh [109]. As to the optimisation and robustness for the measurement of individual node, applying a consensus-based information fusion algorithm is the key to accomplish it. Additionally, consensus schemes, with advantages like scalability, without a sensor fusion centre and no need for global information, make a tremendous contributes to achieving a cooperative fusion over networks. For these reasons, the consensus filtering approach develops rapidly.

2.3.5 K Nearest Neighbors Rule

The nearest neighbour rule is a non-parametric method used for classification and usually regarded as one of the oldest and simplest classifiers [120]. According to the nearest neighbour rule, we can assign a class label to a certain case, if a point that is the closest to the case, based on a set of training data and cases.
The most popular techniques in the distance-based domain of outlier clearance for WSNs are the nearest neighbour-based techniques, which can be employed to derive meaningful information between a specific node and each of its spatially nearest neighbours. This is mainly due to the fact that the nearest neighbour-based techniques can generalize many notions from other techniques and usually make no assumption about sensor measurement distribution [55].

The nearest neighbour rule is a non-parametric method used for classification, which is generally regarded as one of the most mature and straightforward classifiers [120, 121]. Given a set of training data and cases, we can assign a class label to a specific case, if a point is the closest to the case based on the rule.

More specifically, in $k$ Nearest Neighbors (KNN) classification approach, a majority vote of its $k$ spatially nearest neighbours, which are labelled by using the Euclidean distance [122] measure between the test data and training data, can optimally estimate the label of unknown case. There are a set of training data $\varpi_t = (\varpi_{t1}, \varpi_{t2}, \ldots, \varpi_{tn})$, each of which has $\varsigma$ features $(\varpi_{t11}, \varpi_{t12}, \ldots, \varpi_{t1\varsigma})$, and a set of input sample data $\varpi_i = (\varpi_{i1}, \varpi_{i2}, \ldots, \varpi_{im})$ with the same features. The Euclidean distance between $\varpi_{i1}$ and $\varpi_{t1}$ is defined as:

$$\text{Edist}(\varpi_{i1}, \varpi_{t1}) = \sqrt{\sum_{z=1}^{\varsigma} \| \varpi_{i1z} - \varpi_{t1z} \|^2}$$  \hspace{1cm} (2.3)

In [123] and [124], they propose two different distance-based methods to identify global outliers in WSNs. In general, owing to broadcasting the detected outlier to its neighbours for verification and repeating this procedure in each sensor node till the whole WSNs agree on the global outlier, it would cause huge communication overhead in identifying global outliers. The attempt of reducing communication overhead in [124] is over a set of representative sensor information exchanges among neighbour sensor nodes. However, this method still suffers from the communication overhead on account of adopting no network structure, which may lead to a scaling problem for the large scale networks. In addition, Zhang et al. [123] alleviate it through adopting the structure of aggregation tree. But this aggregation tree-based approach may be unstable.
due to the changing of topology of networks.

Hautamaki et al. [125] present a distance-based outlier detection approach using the Indegree Number (ODIN) algorithm, in which a neighbourhood of a point is obtained using a graph. For the given sensor information, ODIN is equal to the number of \( k \) nearest neighbours of the sensor information which has the given sensor information in their \( k \) nearest neighbour list. The inverse of ODIN is the anomaly score for the data instance. A similar method is proposed by [126]. The main drawback of these approaches is the high computational complexity because of the demand for distances of each pair of sensor data. Due to their expensive computational effort, these approaches are not applicable to online operate.

### 2.3.6 Deep Learning

In recent years, the deep learning approaches, including Artificial Neural Networks (ANNs) [127], CNNs [128], etc., achieve the great success in many different fields like computer vision, image classification, speech recognition and drug discovery [129, 130]. Additionally, many works demonstrate that CNNs shows impressive learning performance compared with others, due to its novel structures and methods to extract more abstract and higher-level features for learning.

In fact, CNNs is variants of ANNs which has the simple feed-forward form that it consists of the alternating layers of convolutional operations and pooling operations ending with multiple fully connected layers [131]. Figure 2.1 displays the typical CNNs architectures for image classification. Through Figure 2.1, it is obvious that the major

![Figure 2.1: The very basic CNNs architecture for image classification.](image-url)
change compared with ANNs is the method of joints for their neurons. The neurons connecting pattern with CNNs structure is that only the limited region of input is connected to each neuron instead of the fully connected form. This unique partial connecting pattern, used by CNNs, is capable of exploiting the partial feature information to have spatially contiguous receptive fields so that it can ensure their hidden neurons respond to the spatially partial inputs reliably and strongly. Besides, another important advantage is the shared weights [128] in CNNs which can remarkably decrease the number of weight values, as it further accelerates the learning speed and saves the great computing power for large-scale inputs. Moreover, CNNs utilizes the pooling operation, including max-pooling and average-pooling, to partition the inputs into many different sub-regions. Then, for each sub-region, the maximum value and the average value are obtained over the max-pooling operation and average-pooling, respectively.

Actually, with the rapid development of CNNs, a lot of different efficient CNNs architectures are proposed, such as LeNet [132], AlexNet [128], GoogLeNet [133], etc. Current research works show that, for the time being, CNNs with the deeper and more sophisticated architecture may perform better [134]. For example, the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 competition was GoogLeNet. It achieved a top-5 error rate of 6.67%, which was very close to the human level performance. However, its architecture consisted of a 22 layer deep CNNs. In consideration of the limited resource for each sensor node in networks, we refer to AlexNet as our design prototype which has strong classification ability and usability with less sophisticated architecture, and some modifications are made to design a more appropriate variant to run in WSNs.

As we all know, the deep learning approaches have great success in the fields of image and speech recognition. However, these deep learning methods have rarely been applied in solving classification problems for WSNs. As a matter of fact, sensor measurements in homogeneous multisensor networks have plenty of similarities with an actual grey-scale image on the point of scale of sampling value, the essence of information provided (limitations and complementarities), and redundancy. In this part, we will present these interesting characters of them.
At first, sensor nodes are indeed functionally similar to pixels of an image that both of them denote a particular quantity and provide sampling values. Generally, a grey-scale image involves $m \times n$ pixels describing the same nature information. It is reasonable to consider these pixels as homogeneous multisensor networks consist of $m \times n$ identical optical sensor nodes. And we can regard one of these pixels as a grey-scale intensity sensor providing the intensity information. In the other way round, it is natural that sensor nodes in homogeneous multisensor networks can also be viewed as pixels to measure any specific quantities rather than the intensity information. In the second place, one of the sensor nodes in homogeneous multisensor networks only senses the status of a small sub-region and returning merely a partial measurement due to limited capability in a fixed sampling time. Still, in networks, all the sensor nodes are complementary to each other to provide an utter description of monitored targets. Likewise, the sampling value of one pixel, denoting partial area, is also complementary to these of all the other pixels so that they can compose a particular and complete image. Furthermore, with regard to redundancy, they show the great comparability in that part. In fact, the information of an actual image is always redundant so that it still has a great chance of recognizing the main content in a damaged image, some column vectors of which are lost or mutant. This character is also consistent with that of the sensor measurement in homogeneous multisensor networks for the data redundancy.

### 2.3.7 Deep Reinforcement Learning

In this section, we present the basic knowledge of deep RL and also introduce one of the main categories for deep RL algorithms, the value-based method.

Normally, the machine learning algorithm, RL, is able to accomplish the optimal control of the Markov Decision Process [86]. In applications of RL, there are two main elements: the agent and the environment. The agent takes the action, $a$, within an action space, $A$, to interact with the environment, which evolves over time in a stochastic manner and has a possible state space, $S$, to denote its status. More specifically, in one specific state, $s$, a certain action, $a$, from an agent will result in the reward signal, $r(s, a)$, from the environment to evaluate its behaviour: productive or
ineffective. And also this action will lead to the transition of the environment to the next state.

In this thesis, we mainly focus on the discrete-time case, where agents interact with the environment at each of a sequence of discrete-time steps, \( t = 0, 1, 2, \ldots \). The target of the agent is to learn the optimal policy, \( \pi \), which is able to map states to actions in order to optimize the value function, \( V^\pi(s_0) \), for any initial state \( s_0 \in S \). The value function is designed as:

\[
V^\pi(s_0) = \mathbb{E}_{F_{s_0}}^\pi[G(F_{s_0})],
\]

where \( F_{s_0} \) is a sequence of tuples, \( (s_t, a_t, r_{t+1}) \), \( t \in \{0, 1, 2, \ldots \} \), with \( r_{t+1} = r(s_t, a_t) \) and \( a_t = \pi(s_t) \). Also, \( G(F_{s_0}) = \sum_{t=1}^{\infty} \varrho^{t-1} r_t \) is the discounted total reward with discounted factor, \( \varrho \). In addition, \( Q^\pi(s_0, a_0) \) is the another important function denoting the expected reward under a policy, \( \pi \), with the initial state and action. If the agent follows the optimal policy, \( \pi^* \), the value function and \( Q \) function are presented by \( V^*(s) \) and \( Q^*(s, a) \), respectively. In fact, the Bellman Equation [135] is often used to study the above functions such that they can be rewritten as:

\[
V^*(s_t) = \max_{a_t} [r_{t+1} + \varrho \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) V^*(s_{t+1})]
\]

and

\[
Q^*(s_t, a_t) = r_{t+1} + \varrho \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}),
\]

where \( P(s_{t+1}|s_t, a_t) \) is the transition probability based on the current state and action which is supposed to be obtained by the agent in model-based RL, while model-free RL does not make such assumptions.

Bellman Equation is the fundamental of RL algorithms, which shows the value of the current state is related to the next state and reward. More specifically, as shown in Equation (2.6), the sum of the instant reward and the maximum expected reward thereafter is the current expected reward. In other words, the expected reward for the current state is able to be calculated, once we obtain the future expected reward. In the RL process, firstly, the agent initializes the \( V(s_0) \) or \( Q(s_0, a_0) \) with random values.
Subsequently, it can perform iterative computations involving the policy exploration and policy evaluation phases till the convergence of $V(s)$ or $Q(s,a)$. Additionally, in the policy exploration stage, we adopt one of the soft policies, $\epsilon$-greedy, instead of always keeping the greedy policy to explore the environment so as to learn the optimal policy, where the $\epsilon$ is usually a small value denoting the probability of randomly selecting the action in this policy.

Generally, in some cases with the finite state and action sets, we can store them in the form of matrices. However, with reference to the application of RL algorithms in real-life cases, the size of state sets is huge, sometimes infinite, which means it is impossible to store them in the form of matrices, tables or arrays. To fix the curse of dimensionality problem, Mnih et al. [89] adopt the deep learning concept such that $V(s)$ and $Q(s,a)$ are able to be approximated with a set of parameters over the application of CNNs.

2.4 Summary

In this chapter, the literature review surrounding different information fusion methods which are related to the problems focused by us for IoT systems in this thesis is given firstly, and then the preliminaries about the approaches which are proposed by us in the next chapters are briefly introduced. Based on these materials, typical outlier clearance methods, event classification techniques, and cooperative decision-making approaches for AIoT applications are presented with details about their advantages and disadvantages. Through comparison with the traditional centralized approaches, the advantages of distributed methods are illustrated such that we propose different distributed approaches to solve the problems for IoT systems in this thesis. Also, the shortcomings of those existing distributed methods are discussed. Our proposed methods in the next chapters are focusing on improving the applicabilities and utilities of those existing distributed methods. In other words, these problems in IoT systems and drawbacks of typical methods are our motivations.
Chapter 3

Online Distributed Distance-based Outlier Clearance Approaches for Wireless Sensor Networks

3.1 Introduction

At present, the rapid progress of Micro-Electro-Mechanical Systems technology, sensor technology, embedded computing technology, distributed computation technology, wireless communication technology, etc., has promoted the researches and applications of WSNs forward to a greater and broader scope. Therefore, we are able to witness the successful application and enormous potential of WSNs in multi-disciplinary fields such as target tracking, environmental monitoring, industrial manufacture, smart home, intelligent medical system, traffic control and localisation systems [44,136,137].

A vast quantity of smart sensor nodes comprise WSNs, which are low-power terminal devices with the ability of remote communication and control. In actuality, data acquisition of states of interest and events detection are the main tasks of WSNs. Besides, in most applications of WSNs, it is required to monitor targets and make the event classification based on multisensor fusion results in real-time. However, owing to the imperfect nature of WSNs, the quality of raw sensor measurements collected
by sensor nodes is usually inaccurate and unreliable in practical applications [27]. For one thing, out of cost consideration, a large number of cheap sensor nodes would be employed in networks, the capability and resources of which are stringently limited [138], such as energy supply, internal memory, processing capacity and communication bandwidth, that may lead to collecting questionable and imprecise data. Especially, when the energy supply is out of order or exhausted, the possibility of producing inaccurate data will rise sharply. For another thing, in real applications, numbers of sensor nodes in networks are often randomly or methodically deployed in extremely harsh environments with the enormous complex terrain, which may make some sensor nodes suddenly fail resulting in noise data, spurious data, conflict data, out of sequence data or redundant data [28].

In WSNs, the outlier, which is the sensor node that appears to deviate considerably from other members of the sample or is inconsistent with the remainder of the set of data [139], is one kind of erroneous data that would tremendously adversely affect the quality of multisensor fusion results. From these well-known definitions, it is relatively straightforward to detect and clear outliers by defining a normal behaviour model of sensor measurement to identify outliers, which do not conform to this model. However, in reality, WSNs is often employed to process real-time streaming sensor measurements, which is unbounded and incapable of being modeled precisely, the previous approach would be too laborious to be implemented. Accordingly, a series of critical challenges faced by WSNs is to design a model-free and online outlier clearance approach with low communication overhead, low memory usage, and low computation cost.

In this chapter, we propose two real-time distributed outlier clearance approaches with low resource consumption. In these approaches, we define a weighted average distance-based outlier factor criterion, inspired by the nearest neighbor rule, exploiting sensor measurements from a particular node and its spatially nearest neighbors in the same sample time, to identify outliers in real-time. And also, the false alarm probability of the proposed criterion is derived, which suggests parameter settings in the practical applications. The evaluation performance with the widely used metric of Receiver Operator Characteristic (ROC) curve [140] on the real-life datasets for our proposed approaches and other typical methods shows that the proposed Top-n WAD
and Adaptive Top-n WAD approaches can handle the outliers efficiently and reliably in an online fashion, even if multiple sensor nodes fail at the same given time.

The remainder of the chapter is organized as follows. In Section 3.2, the problem formulation is described. The formal definitions and the related properties of our outlier factor, as well as the details of proposed two online distributed outlier clearance approaches, are explained in Section 3.3 and Section 3.4, respectively. In Section 3.5, the processed results on both synthetic and real-life datasets, using our proposed approaches and the other widely used methods, provide the detailed performance evaluation of all methods. Last, we make the concluding remarks in Section 3.6.

### 3.2 Problem Formulation

Let us consider that a homogeneous multisensor networks monitor a target signal $T(t)$, in which there are $H$ identical time synchronized sensor nodes that are stochastically distorted by noise $W$ and outlier $O$. The noise and the outlier are the zero-mean white Gaussian noise and stochastic erroneous data, respectively. Let a graph, $G = (V, E)$, with its adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N \times N}$, describe the topology of the homogeneous multisensor networks. $V = \{1, ..., H\}$ denotes the sensor nodes, and $E \subseteq V \times V$ represents the set of edges, in which the edge $(v_j, v_i)$ describes the information flow from node $j$ to node $i$. $A$ is defined such that $a_{ii} = 0$, $a_{ij}$ is positive if $(v_j, v_i) \in E$ and $a_{ij} = 0$ otherwise. Note that $a_{ij}$ indicates the weight for the information flow, the edge $(v_j, v_i) \in E$. Then, set $a_{ij} = 1$ when the weights are not relevant, if $(v_j, v_i) \in E$.

Figure 3.1 displays a small multisensor sub-network centered at node $i$, in which they can directly communicate with each other. Let $N_i = \{j \in V : (v_j, v_i) \in E\}$ denote the set of spatially nearest neighbors of sensor node $i$, in-degree of which is also the cardinality of $N_i$, and $J_i = N_i \cup \{i\}$ denote the set of the multisensor sub-network. In addition, set the degree matrix, $D = \text{diag}(d_1, ..., d_H)$, for the homogeneous multisensor networks with $d_i$ as the in-degree number of node $i$. Our objective is to online distributed identify outliers from every new sensor measurement collected by each sensor node in networks and substitute proper estimated data for them.
3.3 Definitions and Properties of the Distance-based Outlier Factor

In this section, firstly, we introduce a series of formal definitions of the weighted average distance-based outlier factor criterion, which is the fundamental of our proposed approaches. We then analyse several properties of the proposed outlier factor criterion, which can provide the guidance for threshold selection in practical applications.

3.3.1 Formal Definitions of WADOF

Definition 3.3.1. Given, $\mathcal{D} = \text{diag}(d_1, ..., d_N)$, the degree matrix of multisensor networks, the nearest neighbours matrix of sensor nodes is:

$$\mathcal{K} := \mathcal{D},$$  \hfill (3.1)

$\mathcal{K} = \text{diag}(k_1, ..., k_i, ..., k_H)$ with $k_i$ as the number of nearest neighbors of sensor node $i$.

Definition 3.3.2. Given the adjacency matrix of multisensor networks, $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$, the set of nearest neighbours of sensor node $i$, i.e., $N_i$, and the measurement
of sensor node, $u$, the weighted average distance of sensor measurement of node $i$ is defined as:

$$\overline{wad}_i := \frac{1}{k_i} \sum_{j \in N_i} \text{dist}(u_i, a_{ij}u_j).$$  \hfill (3.2)

**Definition 3.3.3.** In one sub-network, i.e., $J_i$ ($J_i = N_i \cup \{i\}$), set $A^2_{k_i}$ as the 2-permutation of $k_i$. The internal weighted average distance of sensor measurements in this sub-network, is defined as:

$$WAD_i := \frac{1}{A^2_{k_i}} \sum_{j,l \in N_i; j \neq l} \text{dist}(a_{ij}u_j, a_{il}u_l).$$  \hfill (3.3)

**Definition 3.3.4.** The weighted average distance-based outlier factor of sensor measurement from node $i$ is:

$$WADOF_i := \frac{\overline{wad}_i}{WAD_i}.$$  \hfill (3.4)

**Remark 3.3.1.** In reality, within the sub-network, $J_i$, $WADOF_i$ describes the degree that one certain sensor measurement of node $i$ diverges from those of its $k_i$ spatially nearest neighbours in a specific sampling time. If $WADOF_i \gg 1$, it indicates that the sensor measurement from node $i$ lies outside its sub-network system and it is an obvious outlier. Furthermore, the higher $WADOF_i$ is, the farther sensor measurement of node $i$ is away from those of its $k_i$ spatially nearest neighbours. Oppositely, the measurement of sensor node $i$ is encompassed by those of its neighbours when $WADOF_i \lesssim 1$.

### 3.3.2 Properties of Weighted Average Distance-based Outlier Factor (WADOF)

Undoubtedly, a universal and particular threshold of the proposed outlier factor criterion can offer great convenience to users when distinguishing spurious sensor measurements adopting our criterion in reality. However, the acquisition of the ideal threshold is quite a laborious and time-consuming process due to the sophisticated and diversified application environments and the instability in networks. We can develop an
approaching lower bound of the outlier in sensor measurements, Lower Bound of the Weighted Average Distance-based Outlier Factor (\(LWADOF\)), with few assumptions, though.

**Theorem 3.3.1.** In one sub-network, let \(N_i\) be the set of nearest neighbours of sensor node \(i\), which has \(k_i\) nearest neighbours. For \(k_i \to \infty\), we can obtain the lower bound of outlier:

\[
LWADOF = \frac{1}{2}.
\]  

*Proof.* Let \(u_i \in U = \mathbb{R}\) and \(\bar{u}_i := \frac{1}{k_i} \sum_{m \in N_i} a_{im} u_m\). Using the squared Euclidean distance \(\|\cdot\|^2\), the definitions above can be rewritten as:

\[
\overline{wad}_i = \frac{1}{k_i} \sum_{m \in N_i} \|u_i - a_{im} u_m\|^2
\]

\[
= \frac{1}{k_i} \left[ \sum_{m \in N_i} (a_{im} u_m)^2 + k_i u_i^2 - 2 u_i \sum_{m \in N_i} a_{im} u_m \right]
\]

\[
= \frac{1}{k_i} \left[ \sum_{m \in N_i} (a_{im} u_m)^2 + k_i u_i^2 - 2k_i u_i \bar{u}_i \right]
\]

\[
= \frac{1}{k_i} \left[ \sum_{m \in N_i} (a_{im} u_m)^2 + k_i u_i^2 - 2k_i u_i \bar{u}_i + k_i \bar{u}_i^2 - k_i \bar{u}_i^2 \right]
\]

\[
= \|u_i - \bar{u}_i\|^2 + \frac{1}{k_i} \sum_{m \in N_i} [(a_{im} u_m)^2 - \bar{u}_i^2]
\]

(3.6)
\[ WAD_{i}^{OF} = \frac{1}{A_{k}^{2}} \sum_{m,n \in N_i, m \neq n} \|a_{im}u_m - a_{in}u_n\|^2 \]
\[ \frac{1}{A_{k}^{2}} \sum_{m,n \in N_i, m \neq n} \|a_{im}u_m - a_{in}u_n\|^2 = \frac{1}{k_i(k_i - 1)} \left[ 2(k_i - 1) \sum_{m \in N_i} (a_{im}u_m)^2 - 4(a_{i1}u_{12}u_2 + \cdots + a_{i(k_i-1)}u_{k_i-1}u_{k_i}) \right] \]
\[ = \frac{2}{k_i(k_i - 1)} \left[ k_i \sum_{m \in N_i} (a_{im}u_m)^2 - 2(a_{i1}u_{12}u_2 + \cdots + a_{i(k_i-1)}u_{k_i-1}u_{k_i}) - \sum_{m \in N_i} (a_{im}u_m)^2 \right] \]
\[ = \frac{2}{k_i(k_i - 1)} \left[ k_i \sum_{m \in N_i} (a_{im}u_m)^2 - \left( \sum_{m \in N_i} a_{im}u_m \right)^2 \right] \]
\[ = \frac{2}{k_i(k_i - 1)} \left[ k_i \sum_{m \in N_i} (a_{im}u_m)^2 - (k_i\overline{u_i})^2 \right] \]
\[ = \frac{2}{k_i - 1} \sum_{m \in N_i} [(a_{im}u_m)^2 - \overline{u_i}^2] \]
\[ WADO_{i}^{OF} = \frac{1}{k_i} \sum_{m \in N_i} \|u_i - a_{im}u_m\|^2 \]
\[ \frac{1}{k_i} \sum_{m \in N_i} \|u_i - a_{im}u_m\|^2 = \frac{1}{k_i - 1} \sum_{m \in N_i} \|u_i - \overline{u_i}\|^2 \]
\[ = \frac{2}{k_i - 1} \sum_{m \in N_i} [(a_{im}u_m)^2 - \overline{u_i}^2] \]
\[ \|u_i - \overline{u_i}\|^2 \rightarrow 0 \]

\[ \text{Remark 3.3.2.} \] Theorem 3.3.1 shows that the measurement \( u_i \) is not the outlier, once the \( WADO_{i}^{OF} \approx 0.5 \). The approaching lower bound, \( LWADO_{i}^{OF} \), provides a potential trimming rule, which can considerably reduce the computational complexity in outlier detection proceeding. In practical applications, since \( WADO_{i}^{OF} \) is always positive, one certain sensor measurement can be directly ignored if its \( WADO_{i}^{OF} \) is smaller than \( LWADO_{i}^{OF} \). Only those sensor measurements whose \( WADO_{i}^{OF} \)s are larger than \( LWADO_{i}^{OF} \) are possible to be outliers.

Meanwhile, \( \|u_i - \overline{u_i}\|^2 \rightarrow 0 \), if one is located in the centroid area of its neighbours, which is definitely not an outlier. Taking the ratio, we obtain the lower bound of outlier, \( LWADO_{i}^{OF} \).

\[ \square \]

\[ \text{Remark 3.3.2.} \] Theorem 3.3.1 shows that the measurement \( u_i \) is not the outlier, once the \( WADO_{i}^{OF} \approx 0.5 \). The approaching lower bound, \( LWADO_{i}^{OF} \), provides a potential trimming rule, which can considerably reduce the computational complexity in outlier detection proceeding. In practical applications, since \( WADO_{i}^{OF} \) is always positive, one certain sensor measurement can be directly ignored if its \( WADO_{i}^{OF} \) is smaller than \( LWADO_{i}^{OF} \). Only those sensor measurements whose \( WADO_{i}^{OF} \)s are larger than \( LWADO_{i}^{OF} \) are possible to be outliers.

In real-world applications, it is laborious and time-consuming to achieve proper parameters through the process of trial-and-error. Instead, given the nearest neighbours
matrix and the absolute error of sensor nodes, which is the accuracy of this type of sensor node, we can theoretically ascertain the false detection probability for the specific outlier threshold with some assumptions.

**Theorem 3.3.2.** Assume that, in the homogeneous sensor networks, all sensor observations are independent, identical and distributed, and variance of measurement for each sensor node is $\nu^2$. Given the nearest neighbourhood size of $N_i, k_i$, and the absolute error, $E$, of the specific type sensor node in-service, for the outlier threshold $\epsilon > 1$, the probability of false detecting the sensor measurement of node $i$ as an outlier is:

$$P[WADOF_i \geq \epsilon] \leq \exp\left(-\frac{k_i\nu^4(\epsilon - 1)}{2E^4(1 + 2\epsilon)^2}\right).$$ (3.9)

**Proof.** For simplicity of notation, we assume that networks are not described by a weighted graph, then weights are not relevant and set $a_{ij} = 1$. By means of the computational formula for the variance, we can obtain the expectation of both $wad_i$ and $WAD_i$:

$$\mathbb{E}(wad_i) = \frac{1}{k_i} \mathbb{E}\left(\sum_{m \in N_i} \|u_i - u_m\|^2\right) = 2\nu^2$$ (3.10)

$$\mathbb{E}(WAD_i) = \frac{1}{k_i^2} \mathbb{E}\left(\sum_{m,n \in N_i; m \neq n} \|u_m - u_n\|^2\right) = 2\nu^2$$ (3.11)

We define that $f_T := f_d - \epsilon f_D$, $f_d := \overline{wad}_i$ and $f_D := \overline{WAD}_i$.

As we know that McDiarmid’s Inequality presents the values of a function of i.i.d. variables concentrate around its mean. More exactly, let $f : \mathcal{X}^k \to \mathbb{R}$ satisfies that for all $t = 1, ..., k$ there is a $c_t > 0$ such that for all $x_1, ... x_k, x'_t \in \mathcal{X}$,

$$|f(x_1, \cdots, x_t, \cdots, x_k) - f(x_1, \cdots, x'_t, \cdots, x_k)| \leq c_t$$ (3.12)

then for all $\zeta > 0$,
\[
\begin{align*}
P[f - \mathbb{E}(f) \geq \zeta] &\leq \exp\left(-\frac{2\zeta^2}{\sum_{t \in N_i} c_t^2}\right). \\
(3.13)
\end{align*}
\]

For \( f_d \), we have
\[
\begin{align*}
|f_d(u_1, \ldots, u_d, \ldots, u_{k_i}) - f_d(u_1, \ldots, u'_d, \ldots, u_{k_i})| \\
= \frac{1}{k_i} \left\| u_i - u_d \right\|^2 - \| u_i - u'_d \| \right \| \leq \frac{4E^2}{k_i} = c_d.
(3.14)
\end{align*}
\]

Set \( C^2_{k_i} \) as the 2-combination of \( k_i \) and for \( f_D \), we have
\[
\begin{align*}
|f_D(u_1, \ldots, u_D, \ldots, u_{k_i}) - f_D(u_1, \ldots, u'_D, \ldots, u_{k_i})| \\
= \frac{1}{C^2_{k_i}} \left( \sum_{m \in N_i, m \neq D} \left\| u_m - u_D \right\|^2 - \sum_{n \in N_i, n \neq D} \left\| u_n - u'_D \right\|^2 \right) \right \| \leq \frac{8E^2}{k_i} = c_D.
(3.15)
\end{align*}
\]

For the function \( f_T = f_d - \epsilon f_D \), we have
\[
\begin{align*}
|f_T(u_1, \ldots, u_T, \ldots, u_{k_i}) - f_T(u_1, \ldots, u'_T, \ldots, u_{k_i})| \\
\leq c_d + \epsilon c_D = \frac{4E^2(1 + 2\epsilon)}{k_i} = c_T.
(3.16)
\end{align*}
\]

then, the probability of false alarm is
\[
P[WADOF_i \geq \epsilon] = P[wad_i - \epsilon WAD_i \geq 0] = P[f_T - \mathbb{E}(f_T) \geq \zeta] (3.17)
\]

where \( \zeta = \mathbb{E}(f_T) = 2\lambda^2(\epsilon - 1) \). Using the McDiarmid Inequality, we have
\[
P[WADOF_i \geq \epsilon] \leq \exp\left(-\frac{2\zeta^2}{\sum_{T \in N_i} c_T^2}\right) = \exp\left(-\frac{2\zeta^2}{k_i c_T^2}\right) = \exp\left(-\frac{k_i \lambda^4 (\epsilon - 1)}{2E^4(1 + 2\epsilon)^2}\right). (3.18)
\]

Remark 3.3.3. Theorem 3.3.2 provides the enlightenment for users to select the outlier threshold in practical applications. In actual fact, most sensor measurements with WADOFs less than 1 and greater than 0.5 can be considered as normal data.
noise. In most real applications, users just set certain outlier threshold to clear such obviously unreliable and inaccurate sensor measurements, the WADOFS of which are much greater than 1.

3.4 Online Distributed Outlier Clearance Approaches

The proposed approaches empower every sensor node in networks to exploit sensor data information of itself concerning those of its spatially nearest neighbours, rather than its latest historical data information alone, to identify the new arriving measurements in a real-time fashion.

3.4.1 Top-n WAD

It is extremely onerous to determine a general and specific outlier threshold to identify spurious data in the arbitrary datasets, even though we make the full theoretical analysis in the previous section. Consequently, we adopt the well-accepted Top-n style for each sensor node to discern outliers in its measurement and received measurements from its neighbours by sorting them with larger WADOFS than LWADOFS in the descending order. Then, in each sensor node, the Top-n measurements with the largest WADOFS are regarded as outliers replaced with estimated data, as shown in Algorithm 1, where we label a closed neighbourhood of sensor node $i$ from 1 to $k_i + 1$ for ease of presentation. The sensor nodes are not aware of such local labelling, but each sensor node can identify its nearest neighbours.

3.4.2 Adaptive Top-n WAD

Nevertheless, the main drawback of traditional Top-n style outlier detection is that it could not entirely deal with such cases, in which there are more outliers than pre-defined Top-n value. In real applications, we cannot easily obtain the precise number
Algorithm 1 Top-n WAD runs in sensor node $i$

1: **Input:** $u_1, ..., u_t, ..., u_{k_i+1}$ (measurements from node $i$ and its neighbours)

2: **Initialize** memory variables $N$, $M$, $W$ with zeros

3: $N = [u_1, ..., u_{k_i+1}]$

4: **for** each measurement $u_t \in J_i$ **do**

5: Calculate $WADOF_t$

6: **if** $WADOF_t > LWADOF$ **then**

7: $M = N_t$

8: **else**

9: $W = N_t$

10: **end if**

11: **end for**

12: Sort the first $n$ measurements with the highest $WADOF$ in $M$ and replace by the mean of the remainder measurements

13: **Output:** $u_1P, ..., u_{(k_i+1)P}$ (processed measurements)

of outliers in each step. For the sake of adaptiveness to such uncertainty, we introduce the second approach. In this approach, we employ the sensor measurement with the lowest $WADOF$, which is also smaller than $LWADOF$, as the gauge of the measurement quality. Then, in one step, any measurement with $WADOF$ larger than $LWADOF$ would be regarded as an outlier when its distance from the measurement with the lowest $WADOF$ is greater than twice the absolute error, $E$. However, in extreme cases, new arriving sensor measurements may involve large numbers of outliers, or measurements may be excessively scattered. Thus, the sensor measurement with the lowest $WADOF$ can be greater than $LWADOF$. If such rare cases are encountered, we still incorporate traditional Top-n style into the second approach, which is outlined in Algorithm 2.

Since the fact that outliers account for the minority of the entire dataset, the proposed lower bound saves the computation for the identification of abundant normal sensor measurements in the outlier detection process. Thus, it can greatly save energy consumption to extend the life of networks as far as possible. Besides, the demand for memory usage of each sensor node is relatively compact. Because we only require the
Algorithm 2 Adaptive Top-n WAD runs in sensor node \(i\)

1: **Input:** \(u_1, ..., u_t, ..., u_{k+1}\) (measurements from node \(i\) and its neighbours)

2: **Initialize** memory variables \(N, M, W\) with zeros

3: \(N = [u_1, ..., u_{k+1}]\)

4: for each measurement \(u_t \in J_i\) do

5: Calculate \(WADOF_t\)

6: if \(WADOF_t > LWADOF\) then

7: \(M = N_t\)

8: else

9: \(W = N_t\)

10: end if

11: end for

12: Find the measurement with the lowest \(WADOF, W_{\text{lowest}}\) in \(W\).

13: if \(WADOF_{W_{\text{lowest}}} > 0\) then

14: for each measurement \(u_y \in M\) do

15: if \(\|u_y - W_{\text{lowest}}\| > 2E\) then

16: \(u_y\) is replaced by the mean of normal measurements in \(W\).

17: end if

18: end for

19: else

20: Sort the first \(n\) measurements with the highest \(WADOF\) in \(M\) and replace by the mean of the remainder measurements

21: end if

22: **Output:** \(u_1^P, ..., u_{(k+1)^P}\) (processed measurements)

measurements from its \(k\) neighbours in one sampling time without storing historical measurements. By comparison, SVM-based methods, [40] and [63], require a large amount of historical data to model their own hyper-ellipsoid SVMs. Moreover, supervised and semi-supervised methods, such as [141] and [64], require labelled training sets to create system models. Our approaches, in contrast, can deal with the outlier clearance problem without training sets and long time learning, which are ideal for real-time applications without the prior knowledge about sensor measurements. We
make the Table 3.1 using the Big-O notation [142] to display the comparison of different local outliers detection methods in computational complexity, memory usage and adaptiveness. Calling $U$ the number of nearest neighbours of one sensor node and itself, $b$ the number of features, $sw$ the size of the sliding window used by SVM-based approaches and $lw$ the size of learning window used by the DNOD approach, we have the following approximations as shown in Table 3.1. One should notice that for the sake of high detection accuracy, SVM-based methods and the DNOD approach need to keep the size of their sliding window and learning window relatively large. In other words, $sw$ and $lw$ are much larger than $U$. Moreover, in the interest of the adaptiveness, SVM-based methods need to update their model of targets regularly, which significantly increases the computation complexity. The computational complexity of our approaches mainly depends on two parts. The first is the calculation of $WADOF$, which is straightforward and this step is linear in $U$. The second is that we sort measurements based on their $WADOF$ values, which can be finished in $O(U \log U)$. Since measurements with $WADOF < LWADOF$ are flushed (i.e., they are undoubtedly not outliers), the number of measurements needed to sort in this step is much smaller than $U$. In conclusion, the maximum computation complexity of our approaches is $O(U \log U)$.

<table>
<thead>
<tr>
<th>Methods</th>
<th>References</th>
<th>Model</th>
<th>Identify</th>
<th>Memory</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM-based</td>
<td>[40] [63]</td>
<td>$O(sw^2b + sw^3)$</td>
<td>$O(b)$</td>
<td>$sw$</td>
<td>Yes</td>
</tr>
<tr>
<td>LOF-based</td>
<td>[39]</td>
<td>-</td>
<td>$O(bU^2)$</td>
<td>$U$</td>
<td>No</td>
</tr>
<tr>
<td>DNOD</td>
<td>[62]</td>
<td>-</td>
<td>$O(2bUlw)$</td>
<td>$lw$</td>
<td>Yes</td>
</tr>
<tr>
<td>T-WAD</td>
<td>-</td>
<td>-</td>
<td>$O(bU\log U)$</td>
<td>$U$</td>
<td>No</td>
</tr>
<tr>
<td>AT-WAD</td>
<td>-</td>
<td>-</td>
<td>$O(bU\log U)$</td>
<td>$U$</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 3.5 Synthetic and Experimental Results Analysis

We investigate the performance of proposed approaches in online clearing outliers through tests, in which we adopt other five widely used nearest neighbour-based approaches with Top-n style: LOF [39], KNN [140], INFLO [143], ODIN [125] and RDOS [144] as well as two typical methods: SVM-based EAOD [63] and DNOD [62].
In these comparative tests, we process synthetic sensor measurements and real experimental measurements collected by the SensorScope networks on the EPFL campus [145] and our designed experiments.

The ROC curve is the most popular evaluation metric for outlier detection approaches [140]. A ROC curve is established through plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The TPR is the probability of real outlier detection. And the FPR is the probability of false alarm. The larger the area under the ROC curve, the better the accuracy of the method. Recent years, the metric of ROC curve has been demonstrated to be the most effective measure compared with many others, such as precision and recall, and widely used for the performance evaluation of outlier detection approaches [140].

3.5.1 Synthetic Sensor Measurements Tests

Firstly, in the synthetic data tests, we consider that a closed neighbourhood of the undirected homogeneous multisensor networks, which is centred at a sensor node \( i \) with its 9 spatially nearest neighbours, measure the reference signal with noise and outliers. We use the following test signals:

\[
T(t) = \sin(t + 9) + \sin(2t + 5) + \sin(4t + 2).
\] (3.19)

For all sensor nodes, we set different measurement noise with the same variance, 0.01, and random outliers that there are no more than 4 outliers in the same sampling step. Further, the sampling frequency is 100 Hz and the running time is 4 sec. According to the given noise, we can roughly treat absolute sensor measurement error, \( E \), as 0.1.

Figure 3.2 and Figure 3.3 show the synthetic raw sensor measurements containing the noise only as well as both noise and outliers. Figure 3.4 and Figure 3.5 show the processed results with the Top-2 WAD approach and Adaptive Top-2 WAD approach. Obviously, the Top-2 WAD cannot cope with such cases that there are more than 2 outliers in the same sampling time, while the Adaptive Top-2 WAD can handle those cases, even, in each of which there would be 40% outliers in one step. Moreover, the
TPR and FPR values for the Top-2 WAD are 54.05% and 13.73%, while the TPR and FPR values for the Adaptive Top-2 WAD are 100% and 4.19%.

Figure 3.2: Synthetic data without outliers (T(t)).

Figure 3.3: Synthetic data with noise and outliers (T(t)).

3.5.2 Advanced Synthetic Sensor Measurements Tests

In this section, we compare typical outlier detection methods introduced in [140], such as top-n KNN and top-n LOF, through contrast simulations to investigate the performance of our approach in online clearing outliers in the distributed multisensor fusion
process. In the simulations, we consider that a closed neighbourhood of undirected homogeneous sensor networks, which is centred at a sensor node \( i \) with its 5 nearest neighbours, measure the reference signal with noise and outliers using the consensus-based filter.

We use the following two test signals:
\[ T_1(t) = \sin(t) \]
\[ T_2(t) = \sin(t) + \sin(2t + 3) + \sin(t + 3) \]  
(3.20)

For all sensor nodes, we set different measurement noise with the same variance, 0.04 and random outliers. Further, the sampling frequency is 100 Hz and assume the weights for the communication flow are not relevant. According to the given noise, we can roughly treat absolute sensor measurement error, \( E \), as 1.

**Figures 3.6 and 3.7 show sensor measurements with only noise. Also, the sensor measurements with noise and outliers are presented in Figures 3.8 and 3.9.**

**Figures 3.11, 3.12, 3.13 and 3.14 show sensor fusion results for \( T_1 \) and \( T_2 \), processed with different outlier clearance approaches, when \( n \) is different. Through Figures 3.8 and 3.9, we can see that at some intervals more than two sensor nodes would give**
outliers simultaneously. These traditional methods such as Top-n KNN and Top-n LOF, which are based on the pre-determined number of outliers, \( n \), can not cope with these cases. For instance, three sensor nodes, measuring \( T_1 \), give outliers in 30.7 sec as shown in Figure 3.10. In this case, sensor fusion results, processed with Top-n KNN and Top-n LOF taking pre-determined \( n \) as 1 or 2, are corrupted by undetected outliers according to Figure 3.11 and Figure 3.13. In contrast, the proposed approach can deal with such case adopting our outlier factor criterion. Obviously, one main advantage over other approaches is that our approach can deal with the extreme situations that multiple sensor nodes fail simultaneously in the limited time without prior knowledge about the number of failed nodes.
3.5.3 Real Experiments Sensor Measurements Tests I

The main goal of the experimentations is to collect real data and verify our proposed methods. We design two experiments that one is environment monitoring (temperature
and humidity) with DHT11, and the other is target tracking using multi-webcam Logitech C270. In both experiments, two of our main programs are written in python on Linux Operating System and Raspbian Operating System.

### 3.5.4 Experimental Setup for Tests I

With reference to the first experiment, environment monitoring, Figure 3.15 shows one of smart entities. Each entity consists of one RASPBERRY PI 3 MODEL B and one DHT11 temperature and relative humidity sensor module. Furthermore, RASPBERRY PI 3 MODEL B is the third generation Raspberry Pi, which is primarily responsible for pre-process sensor measurements and logging sensor measurements.

Moreover, DHT11 sensor module can be applied to measure the environmental temperature and relative humidity. We randomly deploy these five sensor nodes in the
office, F1, in Sackville Street Building. The entire monitoring process is last for 2 hours, in which the sampling frequency is 2 Hz.

As for the second experiment, target tracking, the six webcams C270 are used in this experiment. We use python to write the main target tracking program on Linux OS with the help of the OpenCV package. And the multi-webcam tracking system, applied in this experiment with line topology, tracks one green ball-like target. Our tracking system is able to identify one green ball-like target and track its target logging its relative position in csv file. The relative position is the difference in pixels for a specific area between the current frame and the last frame. In our tracking system, there are no reference points, hence it is not able to measure the absolute position. Additionally, the sampling frequency of the tracking system is stable at around 10 Hz. Our fusion strategy involving the WAD and CE filter can improve the robustness of machine vision tracking system. We believe that the idea of our proposed multi-webcam tracking system can be applied in the autopilot system to enhance robust of driving in the harsh environment like heavy rain, sun flash, etc.

Figure 3.16 describes the normal tracking process that the green ball-like target is circled by the yellow solid line, and the centre of the target is marked by a red point.
The lower-left corner of screen shows $dx$ and $dy$, which is the relative position of the centre of target. In addition, the movement tendency of the target would be presented in the upper-left corner of the screen in the camera coordinate system such as West, East, etc.

The main purpose of this experiment is that collect the real sensor measurements, which include the outlier, through this tracking system. Then, we design the interfering action as shown in Figure 3.17, Figure 3.18, Figure 3.19 and Figure 3.20 to obtain the measurements with outlier. Figure 3.17 and Figure 3.18 describe that we use object to transiently block the webcam and target, respectively. Furthermore, as shown in Figure 3.19 and Figure 3.20, we use the flash disturbance to simulate cases that daytime driving with the sun flash and the automobile meeting at night. During the entire process, the target is static, thus, any huge change in relative position would be regarded as the outlier. And also, we set three webcams at the front of the target and the other three at the back of the target due to the limitations of Logitech HD Webcam C270 in wide-angle. And the experiment holds around 200 sec.
3.5.5 Results and Analysis for Tests I

Figure 3.21 and Figure 3.22 show the raw measurements of Celsius Degree and Relative Humidity. It is obvious that these measurements contain outliers, even though our experiment environment is quite stable.

As shown in Figure 3.23 and Figure 3.24, it demonstrates that our proposed approach can clear the outliers with proper estimated data. In fact, it can handle much worse cases, in which outliers occur more frequently, even if the environment of practical application is extremely unstable.

In addition, as presented in Figure 3.25 and Figure 3.26, it is obvious that these measurements are contaminated with outliers, which generated by our disturbances. Because the target is static during the whole experiment.

Figure 3.27 and Figure 3.28 show the fusion results without applied with our proposed approach, obtained through consensus-based filters only. Compare with the fusion
results in Figure 3.29 and Figure 3.30, which are acquired with the proposed approach and consensus-based filters. It is plain to see that our proposed method is able to operate efficiently and reliably with the inaccuracy and fault in our proposed target tracking system, even though multiple webcam sensor nodes fail in the same given time.

3.5.6 Real Experiments Sensor Measurements Tests II

The real experimental sensor measurements are gathered from 97 networked sensing stations deployed on the EPFL campus from July 2006 to May 2007. This measurement campaign aims at better understanding the meteorology and atmospheric transport in the urban environment. Thus, its high temporal and spatial density measurements, covering the macro-homogeneous areas, are extremely suitable for evaluating the performance of outlier clearance approaches.
3.5.7 Experimental Setup for Tests II

In evaluation works, the real-life datasets are collected via a randomly selected cluster of neighbouring nodes centred at node 105 with its 9 spatially nearest neighbours: nodes 12, 13, 14, 15, 18, 24, 39, 44, 45 as illustrated in Figure 3.31. Due to the gigantic amount and multi-attributes of datasets, for the sake of simplification, we randomly choose parts of their measurements, 5380 samples in total, with two attributes: ambient temperature and relative humidity, which are collected by Sensirion SHT75 from 1 am to 4 am on 16th April 2007, avoiding the bias effects on measurements of these two attributes from the shadow and sunlight. Figure 3.32 displays the Sensirion SHT75 and Table 3.2 describes the detailed information of Sensirion SHT75.

<table>
<thead>
<tr>
<th>Table 3.2: Sensirion SHT75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement Type</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Temperature</td>
</tr>
<tr>
<td>Humidity</td>
</tr>
</tbody>
</table>
The original ambient temperature and relative humidity datasets, collected by ten sensor nodes, are shown in Figure 3.33 and Figure 3.35, respectively. More specifically, there are 10 and 20 outliers in total for each attribute, which are thoroughly distinct from others or impossible in reality, and the obvious outliers are labelled. On account of the excessive naïveté of these outlier cases, they can not be adequately used to evaluate the outlier clearance approaches. For this reason, we randomly inject more outliers into these original datasets, which are similar to the outliers existing in the original datasets. Detailed information of the original and injected outlier cases for both datasets are described in Table 3.3 and Table 3.4. Then, Figure 3.34 and Figure 3.36 display the full evaluation datasets, in which there would be more than five outliers in the same step, with the number of outliers, 49 and 46, respectively.
3.5.8 Results and Analysis for Tests II

Figures 3.37, 3.38, 3.39, 3.40, 3.41 and 3.42 show the TPR, FPR and ROC curves on these evaluation datasets obtained via the proposed approaches Top-n WAD and Adaptive Top-n WAD as well as other widely used methods, in which outliers are identified through their outlier detection criteria and then replaced with the mean of remainders. Moreover, the TPR and FPR are produced with different Top-n values ranging from 1 to 5. Perspicuously, the Adaptive Top-n WAD approach exhibits superior outlier clearance performance with the metric of ROC in both datasets. And the Top-n WAD approach shows competitive performance which is slightly better than the other five highly applied methods; meanwhile, the TPR and FPR of these six approaches increase gradually as the Top-n value rising. In addition, even though the Top-n value is 1, the Adaptive Top-n WAD approach can still achieve more than 90 percent TPR and low FPR in diverse datasets containing many outlier cases that
two or more outliers exist at the same sample time, which thoroughly verifies the high adaptability of the Adaptive Top-n WAD. However, under identical circumstance, the other six approaches cannot offer superb outlier clearance performance as the Adaptive Top-n WAD. Particularly, the Adaptive Top-n WAD approach would achieve exceedingly high TPR as almost 100 percent on these datasets and very low FPR at the same time, if we adopt 2 as its Top-n value. As a matter of fact, in real applications, the high Top-n value may obtain the acceptable TPR while it would also lead to the high FPR and information lost owing to the complex unpredictability of the number of outliers in each step. Nevertheless, our Adaptive Top-n WAD approach with the smaller Top-n value cannot only offer high TPR performance but also keep very low FPR and information integrity without prior knowledge of sensor measurements.

Besides, the outlier clearance performances of different typical methods are described by Figure 3.43 and Figure 3.44. In these tests, the Top-n value for proposed approaches and T-LOF is 3. With consideration of the evaluation datasets, we follow the settings
in [63] and [62] for EAOD and DNOD such that the regularisation parameter of EAOD is 0.07 and the size of learning window for DNOD is 100. Obviously, the Adaptive Top-n WAD and SVM-based EAOD achieve remarkable performance and the Top-n WAD approach demonstrates its competitive capabilities in outlier clearance. However, the SVM-based EAOD method requires much higher memory and computing resources, which may cause the outlier detection delay.

3.6 Discussion

In this chapter, we propose two online distributed outlier clearance approaches with low computational complexity and memory usage, in which a weighted average distance-based outlier factor criterion is defined. This criterion is inspired by the nearest neighbor rule, exploiting sensor measurements from a particular node and its spatially
nearest neighbors in the same sample time to identify outliers in real-time. And also, the false alarm probability of the proposed criterion is derived, which suggests parameter settings in the practical applications. It is clear that our proposed methods have the advantages in solving scalability issues, reducing high communication cost, and managing huge memory overhead for the exceptionally large-scale WSNs compared with traditional central type methods [36,37].

Additionally, the evaluation works demonstrate that our approaches achieve better outlier clearance results, even if multiple sensor nodes fail at the same given time, compared with other widely used methods [39, 62, 63, 125, 140, 143, 144]. Moreover, the Table 3.1 display the comparison of different local outliers detection methods [39, 40, 62, 63] in computational complexity, memory usage and adaptiveness. It is obvious that our approaches are better. Particularly, without the prior knowledge of sensor measurements, the Adaptive Top-n WAD approach secures high TPR and keeps low FPR at the same time bringing great convenience to practical applications.
CHAPTER 3. DISTRIBUTED OUTLIER CLEARANCE FOR WSNS

Figure 3.29: Processed relative X data with consensus-based filters and our proposed method.

Figure 3.30: Processed relative Y data with consensus-based filters and our proposed method.

Future works may include the appliance of outlier clearance approaches in multivariate datasets and event classification.
Figure 3.31: The deployment of closed neighbourhood on the EPFL campus.

Table 3.3: Outliers cases for temperature dataset

<table>
<thead>
<tr>
<th>Case</th>
<th>Outliers/Sample</th>
<th>Original Occurrences</th>
<th>After Injection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.4: Outliers cases for relative humidity dataset

<table>
<thead>
<tr>
<th>Case</th>
<th>Outliers/Sample</th>
<th>Original Occurrences</th>
<th>After Injection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
CHAPTER 3. DISTRIBUTED OUTLIER CLEARANCE FOR WSNS

Figure 3.32: Sensirion SHT75 adapted from the Internet.

Figure 3.33: Original temperature datasets (Unit °C).
Figure 3.34: Full evaluation temperature datasets (Unit °C).

Figure 3.35: Original relative humidity datasets.
CHAPTER 3. DISTRIBUTED OUTLIER CLEARANCE FOR WSNS

Figure 3.36: Full evaluation relative humidity datasets.

Figure 3.37: TPR for temperature datasets.
Figure 3.38: FPR for temperature datasets.

Figure 3.39: TPR for relative humidity datasets.
Figure 3.40: FPR for relative humidity datasets.

Figure 3.41: ROC curve for temperature datasets.
CHAPTER 3. DISTRIBUTED OUTLIER CLEARANCE FOR WSNS

Figure 3.42: ROC curve for relative humidity datasets.

Figure 3.43: ROC analysis for different typical methods on temperature datasets.
Figure 3.44: ROC analysis for different typical methods on relative humidity datasets.
Chapter 4

Event Classification in Fixed Topology Wireless Sensor Networks with Distributed 1D-CNNs

4.1 Introduction

Nowadays, as the global accelerating urbanisation, a city faces a series of critical issues including population explosion, deteriorating environment, energy exhaustion, traffic congestion, etc., thus, more and more attention is paid to the future city concept. Sensor networks, in essence, construct the framework of IoT and Big Data in a smart city as the data obtained and analysed by them are fundamental for these services. Further, WSNs are especially important since they consist of inexpensive and compact sensor nodes with low power consumption and maintenance costs which can transmit data remotely and wirelessly as it allows their deployment at an extremely large variety of locations. In fact, WSNs have found an increasingly wide utilisation in different fields for our city including water distribution, pollution management, structural health monitoring, building energy efficiency, intelligent transportation, parking optimisation,
and environmental monitoring [146–148]. Without a doubt, WSNs are the backbone of the future city.

However, for the sake of the superior design and implementation of WSNs in the smart city, we have to deal with significant challenges due to many restrictions posed in such networks. Most constraints mainly stem from the large-scale application of inexpensive sensor nodes with limited resources, such as energy supply and memory storage, on account of the minimum cost principle in the project management. Obviously, the very first and foremost issue is energy availability. Since, one of the main advantages of event-driven sensing technologies is saving great energy, many different event classification techniques, such as in [66–68], are widely applied in WSNs, where sinks transmit the measured data only with the occurrence or the highly possible occurrence of a type or different types of pre-defined events. Then, network centres take related actions based on the decision-making process models. In general, we will define specific events in advance according to the requirements from our users, referred to as the pre-defined events denoting the status of monitored targets. Moreover, the event classification approach allows the efficient management of emergencies and critical incidents owing to the occurrence of certain events, which can lead to the substantial improvement of WSNs in the smart city.

In the previous chapter, we focus on the outlier raging problems in AIoT systems and propose two real-time distributed outlier clearance approaches with competitive features. Once the outliers are removed during the pre-processing phase, the event detection or classification will not be affected. Then, in this chapter, we predominantly focus on a class of distributed deep learning problems in event classification for WSNs. Every sensor node in networks creates its local learning model with its private annotated training data. Then, each sensor node obtains a common global learning model, which is defined as the average of all local ones, via information exchange only within its closed neighbourhood. And our main contribution is proposing a new approach for solving these distributed deep learning problems in WSNs. The proposed method utilizes the powerful deep learning algorithm, CNNs, in combination with consensus theories to enable every sensor node in networks to run local learning and consensus process simultaneously for the sake of learning a common global model,
without using a global data distribution scheme and, hence, requires less information exchange between the sensor nodes, as they just transmit their estimates of the global model instead of exchanging their private annotated training data. Naturally, it greatly reduces the training time and the memory usage of each sensor node, and the communication bandwidths among WSNs. Respective evaluation results indicate that the proposed approach shows the great competitiveness in event classification with compelling features as it eliminates the need for transmitting and synchronizing measurements at central entities and also reduces the average training time and the required memory usage for each sensor node.

The remainder of the chapter is organized as follows. We present the problem layout in Section 4.2. The proposed algorithm is explained in Section 4.3, and also the important component of it, a variant CNNs based on ALexNet [128], is described. In Section 4.4, the proposed algorithm and other typical methods are tested on the structural health monitoring benchmark datasets in order to present the detailed performance evaluation. At last, we make the concluding remarks in Section 4.5.

### 4.2 Problem Formulation

Consider the fixed topology WSNs with $n$ time-synchronized sensor nodes, where each node is capable of measuring the related signals and also individually identifying the corresponding pre-defined events after training. More specifically, our purpose is to learn from the training datasets, which are partitioned and assigned to sensor nodes in networks for distributed learning, such that when detecting a pre-defined event every node can successfully classify it alone, without any cooperative distributed data processing technologies. In addition, for the purpose of saving communication bandwidth in the distributed training phase, it has to be achieved without exchanging any annotated data among sensor nodes. However, in networks, each sensor node can still make the communication with others but not necessarily with every other node, which can be described by a communication graph $G$, where the vertices denote the sensor nodes and the edges denote the communication flows. The graph $G$ shows the neighbour relationships among these $n$ sensor nodes that sensor node $i$ and $j$ are
neighbours if and only if \((i, j)\) is an edge in \(G\). In this chapter, \(N = \{1, 2, ..., n\}\), \(N_i\) and \(E\) indicate the set of sensor nodes, the set of neighbours of sensor node \(i\) and the set of edges of \(G\), respectively. The task of this distributed learning process, in the chapter, is for every sensor node to obtain a common global learning model 
\[ F^* = \left( \frac{\sum_{i=1}^{n} f^*_i}{n} \right), \]
where \(f^*_i\) is the local optimal model to be learned for each sensor node and is the vector in \(\mathbb{R}^m\). The performance of the global learning model, \(F^*\), is equivalent to the centralized case that the single sensor node is trained on the entire training datasets. Obviously, this distributed learning process with partitioned and non-swappable training datasets can remarkably reduce the training time, memory usage and communication bandwidth, which greatly improves the efficiency of WSNs implementation. Also, it does help some applications of WSNs that have problems with collecting the training data in a centralized manner or sharing among sensor nodes as privacy concerns mount and users demand more protections.

In our proposed approach, the distributed learning process is the reiterations of two-phase process. First, the local deep learning, we train each sensor node individually with its private partitioned annotated data using CNNs method. More specifically, this is performed simultaneously at every sensor node that identical sensor nodes are trained locally with different annotated data drawn from the same generating process. Second, the global consensus learning, each sensor node estimates the global learning model with its local information and the information from its neighbours through the consensus algorithm. Finally, the networks can obtain a common global learning model.

### 4.3 Distributed 1D-CNNs Algorithm

As described above, the proposed approach consists of two main process phases. First, we train all sensor nodes in networks, each of which hosts the same 1D-CNNs. Then, the consensus algorithm runs in networks so that the learned quantities are exchanged in related sub-networks while the annotated data is retained locally. It aims to draw the common global learning model as if the entire training datasets are available locally at each sensor node in WSNs.
4.3.1 1D-CNNs for Feature Extraction

Mostly, the conventional CNNs feeds on 2D pixels or 3D voxels, while we use 1D time-series sensor measurements as the input signal, which is fed into the hidden layers, to extract the abstract features in each sensor node. The hidden layers comprise different convolutional layers, pooling layers, and fully connected layers in general. The convolutional layer creates feature maps with convolution filters which are defined by the shared weights and biases linking the input layer to the hidden layer. Afterwards the pooling layer processes the outputs from the convolutional layer in order to reduce the computational complexity and achieve the hierarchical data representation. We apply the max-pooling operation in all pooling layers. In our 1D-CNNs architecture, it uses a few pairs of the convolutional layer and the pooling layer ended with two fully connected layers as well as a softmax output layer [149] to minimize our cross-entropy cost function [150] and bring about the desired output. In addition, for each neuron except those in the softmax output layer, it adopts the Rectified Linear Units (ReLU) function: \( f(x) = \max(0, x) \), as the non-linear activation function instead of the conventional sigmoid function for the sake of a higher rate of convergence and avoiding the vanishing gradient problem [128].

Figure 4.1 describes the basic architecture of our proposed 1D-CNNs with three convolutional layers, three max-pooling layers, two fully-connected layers, and a softmax output layer based on the AlexNet. The main task for our 1D-CNNs is to find a set of appropriate weights and biases to minimize the cost function using feedforward and backpropagation methods. Specifically, it utilizes the feedforward method to obtain the estimated output based on the input signal and the backpropagation method to adjust the weights and biases of neurons in networks by computing the gradient of the cross-entropy cost function. More detailed information about each layer is presented as follows.

In each sensor node, the following equation is able to describe the relationships between the input layer and the convolutional layer:
\[ z_j^l = b_j^l + \sum_{i=1}^{S_{l-1}} 1D\text{conv}(x_i^{l-1}, k_{ij}^{l-1}), \quad (4.1) \]

where \( k_{ij}^{l-1} \) is one of the shared weights from the \( i \)th neuron in layer \( l - 1 \) to the \( j \)th neuron in the layer \( l \), and \( x_i^{l-1}, b_j^l, z_j^l \) and \( S_{l-1} \) denote the input of the convolutional layer, the bias of the \( j \)th neuron in layer \( l \), the weighted input to the \( j \)th neuron in layer \( l \), and the size of the shared weights in layer \( l - 1 \), respectively. Then, after the convolutional operation, \( 1D\text{conv}(\cdot) \), each neuron in layer \( l \) will generate the value, \( x^l \), for the next max-pooling layer using the ReLU activation function with the weighted input information, \( z^l \), as:

\[ x^l = \text{ReLU}(z^l). \quad (4.2) \]

Next, the output of the max-pooling layer can be expressed as:

\[ x^{l+1} = \text{MaxP}(x^l), \quad (4.3) \]

where \( \text{MaxP}(\cdot) \) denotes the max-pooling operation, and \( x^{l+1} \) is the input for the next
layer. Afterwards the fully-connected layers, following three different pairs of convolutional layer and pooling layer, connect every neuron from the max-pooling layer to every one of the neurons in the softmax layer, which is also our output layer. The number of output neurons, \( M \), is related to the number of pre-defined events. In the output layer, the softmax function is our activation function which will generate the probability distribution for pre-defined events. More specifically, the estimated probability for the weighted input of the \( j \)th output neuron in the final layer, \( z_j^L \), is

\[
P(z_j^L) = \frac{e^{z_j^L}}{\sum_M e^{z_k^L}}.
\] (4.4)

Eventually, the estimated event label, \( y \), is obtained. And the corresponding cross-entropy cost function can be computed as followed:

\[
C = -\sum_h \sum_r (Y_{hr} \log y_{hr}),
\] (4.5)

where \( r \) is the number of events in the \( h \)th training set, and \( Y_{hr} \) is the actual event label.

Altogether, the deep network is designed and trained to minimise the cross-entropy cost function (4.5). To prompt the rapid convergence, we utilize the Stochastic Gradient Descent (SGD) algorithm and a robust weight initialisation method [151], that particularly considers the rectifier nonlinearities, and also set the appropriate learning rate, \( \eta \), in the training process. Then, it can update the weights and biases in all the convolutional layers and fully-connected layers as follows:

\[
k_{ij}^{l-1}(t+1) = k_{ij}^{l-1}(t) - \eta \frac{\partial C}{\partial k_{ij}^{l-1}},
\]
\[
\theta_j^l(t+1) = \theta_j^l(t) - \eta \frac{\partial C}{\partial \theta_j^l}.
\] (4.6)

### 4.3.2 Distributed 1D-CNNs Process

The local learning and global learning are combined to perform at each sensor node \( i \) to obtain its local learning model \( f_i \) and the estimated global learning model \( F_i \).
At the beginning time $t = 0$, each sensor node $i$ sets $F_i(0) = f_i(0)$. Then, for each time $t > 0$, every sensor node $i$ repeats the following steps in order as shown in Algorithm 3, where $d_i$ is the cardinality of $N_i$. Furthermore, the neighbours of sensor node $i$ are labelled from 1 to $d_i$ for the purpose of plain presentation. As described in problem layout, in this chapter we consider the fixed topology networks, and thus $d_i$ is not changed. When setting the constant edge weights [152] in Algorithm 3, it can acquire the best possible ones with

$$
\gamma = \frac{2}{\lambda_1(L) + \lambda_{n-1}(L)},
$$

(4.7)

where $L$ is the Laplacian matrix of graph $\mathcal{G}$, and $\lambda_i(\cdot)$ denotes the $i$th largest eigenvalue of a symmetric matrix. At last, we terminate the algorithm when the stopping function, $S(\Upsilon, \psi)$, is activated, where $\Upsilon$ and $\psi$ represent the maximum number of iterations and the user-specified threshold $\psi \in \mathbb{R}^+$, respectively. Moreover, the stopping function is activated if the classification error of the estimated global learning model at each sensor node is less than $\psi$, or the number of iterations reaches to $\Upsilon$.

**Remark 4.3.1.** The weight matrix, $W = [w_{ij}]$, with constant edge weights is a symmetric doubly stochastic matrix used for the distributed linear averaging problem.

In fact, to make our main result justified, it is clearly necessary that every sensor node, $i$, can be trained locally to learn $f_i^*$ with 1D-CNNs, and also the graph of networks must be connected. Therefore, we impose the following assumptions in this chapter.

**Assumption 4.3.1.** For every sensor node, $i$, in networks, the deep learning algorithm, 1D-CNNs, guarantees that the local learning model, $f_i(t)$, converges to $f_i^*$.

**Assumption 4.3.2.** The graph, $\mathcal{G}$, of WSNs is connected.

**Theorem 4.3.1.** Suppose that Assumptions are hold. In the fixed topology WSNs with the connected graph, $\mathcal{G}$, if all $n$ sensor nodes follow the Algorithm 3, then:

$$
\lim_{t \to \infty} F_i(t) = F^*, i \in \mathcal{N}.
$$

(4.11)
Algorithm 3 Distributed 1D-CNNs in Sensor Node $i$

1: Start Time: $t$

2: **Input from its neighbours**: Sensor node $i$ receives the estimated global learning models, $F_1(t),...,F_d(t)$, from its neighbors.

3: **Local learning with 1D-CNNs**: Sensor node $i$ obtains its new local learning model, $f_i(t+1)$, with the 1D-CNNs and also computes $e_i(t)$ as follows:

$$e_i(t) = f_i(t+1) - f_i(t). \quad (4.8)$$

4: **Global learning with consensus**: Sensor node $i$ updates its new estimated global learning model, $F_i(t+1)$, through

$$F_i(t+1) = w_{ii}F_i(t) + \sum_{j\in N_i} w_{ij}F_j(t) + e_i(t), \quad (4.9)$$

where $w_{ii}$ and $w_{ij}$ are constant edge weights defined as follows:

$$w_{ij} = \begin{cases} 
\gamma & \text{if } (i,j) \in \mathcal{E}, \\
1 - \gamma d_i & \text{if } i = j, \\
0 & \text{otherwise.} 
\end{cases} \quad (4.10)$$

5: **Terminate operation**: Sensor node $i$ computes the stopping function $S(\Upsilon, \psi)$ in order to terminate the algorithm or continue.

6: **Output to its neighbours**: Sensor node $i$ sends its new estimated global learning model, $F_i(t+1)$, to its neighbours.

7: **Next Time**: $t+1$

### 4.3.3 Convergence Analysis

Firstly, consider the solution for the distributed average consensus problem in the normal sensor networks system, which is described in [153], i.e.,

$$F_i(t+1) = w_{ii}F_i(t) + \sum_{j\in N_i} w_{ij}F_j(t), \quad i = 1, 2, 3, \ldots, n. \quad (4.12)$$

We use the distributed linear iterative approach, equation (4.12), to obtain the arithmetic mean of states of sensor nodes. And each sensor node updates its state with a linear combination of its own information and the information from its neighbours.

When the state information is $m$ dimensional, the equation (4.12) can be written in
the new form, i.e.,

$$F(t + 1) = (W \otimes I)F(t),$$  \hspace{1cm} (4.13)

the convergence analysis of which has been well studied in [152,153].

**Proposition 4.3.1.** [153] In the fixed topology WSNs, if its graph $G$ is connected, and the corresponding weight matrix is paracontracting with respect to the Euclidean norm, then, system (4.13) achieves a consensus:

$$\lim_{t \to \infty} F(t) = \lim_{t \to \infty} W^t F(0) = \frac{11^\top}{n} F(0),$$  \hspace{1cm} (4.14)

for all $F(0) \in \mathbb{R}^{nm}$.

Due to that $F_i(t)$ and $f_i(t)$ are vectors in $\mathbb{R}^m$, and WSNs consist of $n$ nodes, the equation (4.9) can be written in the form of an $nm$-dimensional state equation as following:

$$F(t + 1) = (W \otimes I)F(t) + f(t + 1) - f(t) = (W \otimes I)F(t) + e(t),$$  \hspace{1cm} (4.15)

where $F(t) = [F_1(t), F_2(t), F_3(t), \ldots, F_{n-1}(t), F_n(t)]^\top$, $f(t) = [f_1(t), f_2(t), f_3(t), \ldots, f_{n-1}(t), f_n(t)]^\top$, and $e(t) = [e_1(t), e_2(t), e_3(t), \ldots, e_{n-1}(t), e_n(t)]^\top$.

Accordingly, compared with system (4.13), system (4.15) can be regarded as a distributed consensus process with a time-dependent disturbance, $e(t)$.

**Lemma 4.3.1.** For any connected graph, the constant edge weight matrix is paracontracting with respect to the Euclidean norm.

**Proof.** It is clear that the constant edge weight matrix is the symmetric stochastic matrix based on the definition (4.10). As a result, it is the valid transition probability matrix for a Markov chain on the connected graph [154, 155], which means that its eigenvalues are all real and lie in the interval [-1,1].

In addition, the definition (4.10) ensures that all the diagonal entries, $w_{ii}$, are strictly positive. In other words, the related Markov chains are aperiodic. Hence, the constant
edge weight matrix is paracontracting with respect to the Euclidean norm, all the eigenvalues of which lie in the interval \((-1,1]\).

\[\Box\]

**Lemma 4.3.2.** If the sequence matrices, \(W^t\), is uniformly ergodic and the Assumption 4.3.1 is hold, then all \(F_i(t)\) in system (4.15) will asymptotically achieve a consensus:

\[
\lim_{t \to \infty} \|F_i(t) - F_j(t)\| = 0 \quad (4.16)
\]

for all \(i, j \in \{1, 2, 3, \ldots, n\}\).

Proposition 4.3.1 is the direct consequence of this lemma once each local learning model reaches its \(f_i^*\).

**Proof of Theorem 1.** Under Assumption 4.3.2, in the fixed topology WSNs with the connected graph, \(G\), suppose that all \(n\) sensor nodes adhere to Algorithm 3, then, \(W\) is the paracontracting matrix according to Lemma 4.3.1. Further, it is known that if \(W\) is the paracontracting matrix and the corresponding graph is connected, the sequence matrices, \(W^t\), is uniformly ergodic [153]. Consequently, based on Lemma 4.3.2, all \(F_i(t)\) will asymptotically achieve a consensus with Assumption 4.3.1.

For each sensor node, \(i\), we set that \(F_i(0) = f_i(0)\) at the beginning time \(t = 0\). Then,

\[
\sum_{i=1}^{n} F_i(0) = \sum_{i=1}^{n} f_i(0). \quad (4.17)
\]

In the next part of proof, the induction method on \(t\) will be utilized. For \(t = \sigma\), where \(\sigma \in \mathbb{Z}^+\), it supposes \(\sum_{i=1}^{n} F_i(\sigma) = \sum_{i=1}^{n} f_i(\sigma)\), which means that

\[
(1 \otimes I)^\top F(\sigma) = (1 \otimes I)^\top f(\sigma). \quad (4.18)
\]

As for system (4.15), we have
\[(1 \otimes I)^T F(\sigma + 1) = (1 \otimes I)^T (W \otimes I) F(\sigma) + (1 \otimes I)^T e(\sigma) \]
\[= (1 \otimes I)^T (W \otimes I) F(\sigma) + (1 \otimes I)^T f(\sigma + 1) \]
\[= (1^T \otimes I^T) (W \otimes I) F(\sigma) - (1 \otimes I)^T f(\sigma) \]
\[+ (1 \otimes I)^T f(\sigma + 1) \]
\[= (1^T W) \otimes (I^T I) F(\sigma) - (1 \otimes I)^T f(\sigma) \]
\[+ (1 \otimes I)^T f(\sigma + 1) \]
\[= (1 \otimes I)^T F(\sigma) - (1 \otimes I)^T f(\sigma) \]
\[+ (1 \otimes I)^T f(\sigma + 1) \]
\[= (1 \otimes I)^T f(\sigma + 1), \tag{4.19} \]

which implies that \(\sum_{i=1}^{n} F_i(t) = \sum_{i=1}^{n} f_i(t)\) holds for \(t = \sigma = 1\). Therefore, by induction, we have

\[\sum_{i=1}^{n} F_i(t) = \sum_{i=1}^{n} f_i(t) \tag{4.20} \]

for all \(t \geq 1\).

Thus, the system (4.15) will achieve a consensus and equation (4.20) corrects at all time, if all sensor nodes in the fixed topology WSNs adhere to Algorithm 3 with two Assumptions.

Next, with the Assumption 4.3.1, it is clear that

\[\lim_{t \to \infty} \sum_{i=1}^{n} F_i(t) = \lim_{t \to \infty} \sum_{i=1}^{n} f_i(t) = \sum_{i=1}^{n} f_i^*. \tag{4.21} \]

Since the system (4.15) will achieves a consensus, we have

\[\lim_{t \to \infty} F_i(t) = \frac{1}{n} \sum_{i=1}^{n} f_i^* = F^* \tag{4.22} \]
for all $i \in N$.

\section*{4.4 Evaluation Results and Analysis on Real-life Datasets}

In this section, we investigate the performance of our proposed algorithm, distributed 1D-CNNs, and other typical approaches with the real experimental datasets, which are from the Qatar University Grandstand Simulator [156]. This simulator, as shown in Figure 4.2, is configured with a great number of sensor nodes, the labels of which are displayed by the ROW number (1, 2, 3, 4, 5) and COLUMN number (1, 2, 3, 4, 5, 6), on its main girders. It can be used to verify the machine learning-based event classification algorithms (structural damage detection). It is clear to see that 8 main girders and 25 filler beams form the grandstand simulator with footprint dimensions of $4.2 \text{ m} \times 4.2 \text{ m}$. And also there are 30 uniaxial accelerometers (vertical direction) in total installed at its 30 joints. The labels of entities are displayed by the ROW number (1, 2, 3, 4, 5) and COLUMN number (1, 2, 3, 4, 5, 6).

More specifically, the real experimental datasets are generated from 30 different structural damage scenarios simulated by loosening the bolts at the beam-to-girder connections. In the 30 different structural damage scenarios, the damage is introduced to each of the 30 joints, respectively, and the joint numbers are shown in Figure 4.2 in matrix form. Additionally, in each damage scenario, the sampling frequency for every accelerometer is 1024 Hz, and the recording time is 256 s.

\subsection*{4.4.1 Experimental Setup}

In the evaluation works, 12 sensor nodes with different positions, which are marked with red ticks, are chosen to create the networks. Then, we utilize the proposed algorithm, distributed 1D-CNNs, to train the networks with part of the real experimental
datasets and treat the rest as the test datasets. The variant CNNs based on the AlexNet is designed to achieve a compact structure with three pairs of convolutional layer and pooling layer ended with two fully connected layers and a softmax output layer. Such a design strategy is intended for maximizing the utilisation efficiency of WSNs resources. Furthermore, the following evaluation results imply that it may be completely unnecessary to achieve the desired classification performance in WSNs with deep and sophisticated structure CNNs.

In fact, the final structure and all the parameters of 1D-CNNs are acquired via trial-and-error that it has 36 feature maps, each of which is defined by the shared weights with size = 41, stride = 1, and non-padding as well as a single shared bias, and all the convolutional layers adopt the same size of shared weights and bias. As for max-pooling operations, they are all set with size = 2 and stride = 2. Also, there are 30 neurons in each fully connected layer, and the size of the softmax output layer is 2, which is related to the number of classes: Undamaged case and Damage case. Moreover, in every structural damage scenario, each signal collected by the accelerometer contains 262144 samples, and we group, divide into frames, balance and normalize the datasets.
based on the rules in [156]. However, each used frame length is 1024 and 80% of these frames are utilized, which are evenly partitioned into 12 datasets and assigned to the networks with the simple fixed undirected communication topology as shown in Figure 4.3, while the remaining 20% are regarded as test datasets for the distributed training process. Also, the $\Upsilon$ and $\psi$ for the stopping function, $S(\Upsilon, \psi)$, are set to 500 and 1%, respectively. As mentioned in section 4.3, for the sake of better training performance, we apply the robust weight initialisation method [151], and the optimisation algorithm, SGD, and set the learning rate as 0.001 in the distributed training process.

![Figure 4.3: The communication topology of sensor networks.](image)

### 4.4.2 Results and Analysis

The classification performance of the distributed trained networks will be tested against 12 structural damage cases, where case 1 to 12 correspond to loosening one single joint of the selected joints marked with red ticks as shown in Figure 4.2, such as Case 1: Joint (1,1) loosened; Case 2: Joint (1,2) loosened; Case 3: Joint (1,3) loosened; Case 4: Joint (1,4) loosened; Case 5: Joint (2,1) loosened; ...; Case 12: Joint (3,4) loosened. In each case, we feed related test signals, which are from test datasets and normalized as consecutive frames, to 12 distributed trained sensor nodes with the aim of obtaining the Probability of Damage (PRoD) of joints. PRoD$_{ij}$ denotes the proportion of detected damage frames in total test frames for joint $(i,j)$. Besides, the centralized case
and another typical distributed learning framework of CONS-RVFL [42] are tested, respectively. In the centralized case, one sensor node is trained with all training datasets using the identical 1D-CNNs, with the same test datasets. Also, we follow the setting in [42] to test the framework of CONS-RVFL. Figure 4.4, Figure 4.5 and Figure 4.6 show the PRoD distributions for the 12 cases using corresponding approaches, respectively.

Figure 4.4: The PRoD distributions for 12 damage cases when adopting CONS-RVFL framework based 1D-CNNs.

Generally, the evaluation results illustrate that the proposed distributed 1D-CNNs can achieve the success in monitoring the healthy condition of the grandstand and detecting the loosened joints, as the outputs, PRoD values, plainly show the location of damage in all 12 different structural damage scenarios. From Cases 1 to 12, the trained networks have successfully allocated low PRoD values to the undamaged joints and higher PRoD values to the damages. The PRoD distributions displayed in Figure 4.5 can be regarded as the evident success, in consideration of the fact that this is
Figure 4.5: The PRoD distributions for 12 damage cases when adopting proposed distributed 1D-CNNs.

the pioneer application of distributed 1D-CNNs in sensor networks. The framework of CONS-RVFL has also been proven capable of detecting damage cases. However, the overall performance is not as good as our proposed approach. There are several matters to be considered in the implementation of CONS-RVFL framework for distributed learning. First, no node is able to learn its local model accurately within a finite number of iterations. The consensus process needs to be run again if each node repeatedly updates its local model. Also, the termination function may cause nodes to stop asynchronously. In each loop, unless the consensus process has converged, the overall learning model may be different among nodes [41]. Such matters may mainly lead to negative effects on learning performance. The proposed approach is designed to run the local learning and consensus process simultaneously to get around the above limitations.
CHAPTER 4. EVENT CLASSIFICATION IN WSNS

Figure 4.6: The PRoD distributions for 12 damage cases when adopting centralized 1D-CNNs.

Furthermore, the results of evaluation works demonstrate that our proposed distributed 1D-CNNs is capable of tracking the centralized 1D-CNNs very efficiently, albeit the centralized solution holds a slender lead in some cases. However, the distributed 1D-CNNs takes a great advantage in training time and memory usage compared to the centralized solution. Figure 4.7 and Figure 4.8 present the percentage of training time and required memory usage for labelled training data of the centralized solution for one node in networks with different scales on the same training datasets. It is obvious that the distributed 1D-CNNs requires noticeably low training time and memory usage for labelled training data compared with the centralized 1D-CNNs, as much as an order of magnitude for considerably large networks. Also, it can make great contributions to some cases that collecting the training data in a centralized manner for networks and sharing training data among networks are very expensive and cumbersome, or even impossible as privacy concerns mount and users demand...
Figure 4.7: The training time analysis for different nodes of networks.

Figure 4.8: The memory usage for labeled training data analysis of different nodes of networks.

more protections.
4.5 Discussion

In the previous chapter, two real-time distributed outlier clearance approaches with competitive features are proposed to solve the outlier raging problems in AIoT systems. Once the outliers are removed during the pre-processing phase, the event detection or classification will not be affected. Then, in this chapter, a solution combining deep learning technologies with consensus theories is proposed to a class of distributed deep learning problems in event classification for WSNs. It allows that each sensor node in networks establishes the local model with private training datasets so as to obtain a common model via communication only among its neighbours. More specifically, the proposed method utilizes the powerful deep learning algorithm, CNNs, in combination with consensus theories to enable every sensor node in networks to run local learning and consensus process simultaneously for the sake of obtaining a common global learning model, without adopting a global data distribution scheme. Also, it requires less information exchange between the sensor nodes compared with centralized solutions [30, 31], as sensor nodes just transmit their estimates of the global learning model instead of exchanging their private annotated training data. Naturally, it greatly reduces the training time and the memory usage of each sensor node, and the communication bandwidths among WSNs.

In addition, the verification of our proposed approach and other typical methods [41–43] on the structural health monitoring benchmark datasets shows that the proposed approach is capable of extracting the abstract features of specific events in order to indicate the health of monitoring structure with high accuracy, efficiency and reliability. Besides, the proposed learning framework is fully decentralized in the training phase and event classification phase so that it eliminates the need for transmitting and synchronizing measurements at central entities and also reduces the average training time and the required memory usage for each sensor node compared with centralized methods. Beyond question, the proposed approach is able to make a great contribution to the construction of a smart city. With respect to future work, it is of interest to study the effect of networks topology on the event classification performance in
WSNs. And also we expect to solve the switching topology and communication time-delay problems in the training phase for WSNs.
Chapter 5

Coordinated Sensing Coverage in Wireless Sensor and Actuator Networks with Distributed Deep Reinforcement Learning

5.1 Introduction

In the previous chapters, the shortcomings and drawbacks for traditional outlier clearance approaches and learning frameworks in networks are presented. In order to solve such problems, two innovative distributed outlier clearance approaches and a novel distributed learning framework are proposed. We also present the analysis of these proposed methods and the evaluation works in data cleaning and event classifications to demonstrate that our approaches are reliable and effective in the real IoT applications.

Once the accurate data and information are obtained, the decision making becomes the final key part in the AIoT systems. As a matter of fact, the proposed distributed learning framework is able to be applied in the distributed decision making. Next, we are going to focus on the problems in decision making for WSANs.
In the last decades, as the cost of bandwidth, computing, memory, sensors, etc. has fallen dramatically around the world, we have witnessed the formation of the early stage of the IoT, which was getting devices connected and making them smart. However, in the wake of developments in both analytic methodology and artificial intelligence, we are right on the edge of the next wave of IoT, which is about achieving the autonomy for those smart devices to be capable of sensing, understanding, adapting and reacting to the physical world. In other words, IoT devices turn into AIoT through integrating those devices and machine learning algorithms. The application of AIoT in the real world is beginning to take shape, such as vehicular computing and networks in intelligent transportation [5–9], energy trading and management in smart grid [10–13], and communication and energy consumption optimisation in WSANs [14–16,157].

Generally, the AIoT networks have the dynamic perception of the external environment and make the related control decisions to react. WSANs, in essence, are the perfect implementation environment of the AIoT concept, where the AIoT devices with sensors and actuators are deployed. WSANs are a large number of low-cost wireless devices with the ability to real-time sense the physical world (events) and rapidly execute the relatively complicated actions (solutions), based on the collected, shared and fused data by smart devices. More specifically, after deployed, sensor nodes are in charge of accumulating the information of interest, carrying out local processing of these data including information cleaning, compression, broadcasting, fusion, analysis, etc. and generating control decisions for the actuators to react. The actuators, on the other hand, may have a simple mechanical structure while possessing sufficient computing power and energy resources so as to perform complex tasks such as valve operation, turntable control, gearbox shift, etc. In fact, WSANs have found an increasingly wide utilisation in different fields such as pollution management, building automation, smart grid, intelligent transportation, etc. [17–20,158]. Notwithstanding, we must handle many critical issues arising in WSANs such as the network delay, sensing coverage problem, energy consumption optimisation, etc., for the sake of the superior design and implementation of WSANs. And the sensing coverage optimisation is one of the most principal problems addressed in WSANs literature [32–35]. In this problem, the main task for networks is to appropriately cover the target environment so as to guarantee
that all pre-defined events which occur in that area can be reliably, accurately and efficiently identified by networks with minimum energy consumption.

In the previous chapter, we focus on a class of distributed learning problems in event classification for networks and propose a novel distributed learning framework with compelling features. After the limited resources problems and privacy issues in event classification stage for AIoT systems are solved, the user or smart agent can make the sound decision. Then, in this chapter, we predominantly focus on a class of distributed deep RL problems in sensing coverage optimisation for WSANs. Following the proposed algorithm, each entity in networks updates its local RL model with its private experience replay memory and then obtains a common global model with consensus theories that information exchange is only within its closed neighbourhood. At last, all the entities are able to learn the optimal policy. Similar works can be found in [41–43]. Nevertheless, all the above distributed learning algorithms may be subject to the asynchronous termination that the stop criteria at some entities may be valid for termination but other entities have not achieved the same decision. Additionally, it is obvious that once they execute more consensus steps altogether the convergence rate of them will be quite slow as they comprise two successive processes where the consensus process is slow compared with the local learning. And our main contribution is proposing a new approach that it performs the learning process and consensus process simultaneously to solve the above limitations.

The rest of the chapter is organized as follows. In Section 5.2, the problem formulation is described. In Section 5.3, we present an overview of the proposed algorithm. In Section 5.4, the evaluation works are designed to verify the proposed algorithm so as to show the detailed performance evaluation. At last, we make the concluding remarks in Section 5.5.

5.2 Problem Formulation

Consider the target environment represented by a 2-dimensional 20 × 20 grid. We randomly distribute 6 smart entities on this area with the label from E1 to E6 which
possess the sensing and executive capabilities. Specifically, with regard to the discrete action space for them, each entity has a finite action set including the Normal Mode, Low-Power Mode, and Standby Mode. The entity in the Normal Mode is able to cover $81 \ (9 \times 9)$ blocks around itself while the entity in the Low-Power Mode can cover $49 \ (7 \times 7)$ blocks. Also, the Standby Mode will cut down the most power of the entity and only keep the sufficient power supply for the response to the wake-up events. Besides, each entity is able to identify the status of blocks $(9 \times 9)$ around itself in any mode that the blocks are covered or not covered by any entities, even partially. The aim of all entities in networks is learning to take the optimal actions to collaborate with each other such that they are able to perform the best level of sensing coverage in the most energy-efficient way. Additionally, in networks, each entity is able to make the communication with others that the communication topology can be illustrated by a fixed graph, $G$, where the vertices represent the entities, and the edges represent the communication flows. The fixed graph, $G$, describes the neighbour relationships among these entities that entity $i$ and $j$ are neighbours if and only if $(i, j)$ is an edge in $G$. In this chapter, $\mathcal{N} = \{1, 2, ..., 6\}$, $\mathcal{N}_i$ and $\mathcal{E}$ indicate the set of entities, the set of neighbours of entity $i$ and the set of edges of $G$, respectively.

5.3 Distributed Deep Reinforcement Learning for Coordinated Sensing Coverage

As described above, the proposed approach consists of three main phases. Firstly, we leave all entities in networks to randomly interact with the environment, each of which hosts the RL algorithm, so as to generate the local experience Q-values and store them in the experience replay memory. Then, the deep learning algorithm, CNNs, and consensus algorithm run simultaneously in networks such that the learned quantities are exchanged in corresponding sub-networks while the annotated Q-values are retained locally. It aims to draw the common global model as if all the experience replays from entities are integrated available locally at each entity in networks.
5.3.1 Distributed Deep Reinforcement Learning Process

As described in problem formulation, the state of each entity is similar to the binary image, which is able to be presented with 0 and 1. And the entity is able to use Deep Q-Network [89] to obtain its local model. Before the implementation of distributed learning, each entity needs to interact with the environment independently for the accumulation of the experience replay. Then, the local learning and global learning are combined to run at each entity \( i \) to obtain its local learning model \( f_i^* \) (\( f_i^* \) is the local optimal model) and the estimated global learning model \( F_i \) (\( F^* \) is the common global learning model).

We set \( F_i(0) = f_i(0) \) for each entity \( i \) at the start time of distributed learning. Next, along with \( t > 0 \), each entity \( i \) performs the Algorithm 4, where \( d_i \) is the cardinality of \( N_i \). Moreover, for the purpose of plain presentation, we label the neighbours of entity \( i \) from 1 to \( d_i \). In this chapter, we only consider the fixed topology networks so that \( d_i \) is unchanged. Also, it can be obtained the best constant edge weights [152] in Algorithm 4 with

\[
\varphi = \frac{2}{\mu_1(\mathcal{L}) + \mu_{n-1}(\mathcal{L})},
\]

where \( \mathcal{L} \) is the Laplacian matrix of graph, \( \mathcal{G} \), and \( \mu_i(\cdot) \) represents the \( i \)th largest eigenvalue of a symmetric matrix. Finally, the algorithm is terminated if the stopping function, \( S(\Gamma) \), is activated, where \( \Gamma \) denotes the maximum number of iterations.

**Remark 5.3.1.** \( W = [w_{ij}] \) with constant edge weights is a symmetric doubly stochastic matrix used for the distributed linear averaging problem.

Actually, it is necessary that each entity \( i \) can be trained locally to learn \( f_i^* \), and also the communication topology must be connected so as to make our results justified. As a result, the following assumptions are imposed through this chapter.

**Assumption 5.3.1.** For each entity \( i \) in networks, the local learning model, \( f_i(t) \), will converge to \( f_i^* \).

**Assumption 5.3.2.** The communication topology, \( \mathcal{G} \), is connected.
Algorithm 4 Distributed Learning at Entity $i$

1: **Start Time:** $t$

2: **Input from its neighbours:** Entity $i$ obtains the estimated global learning models, $F_1(t), \ldots, F_d(t)$, from its neighbours.

3: **Local learning:** Entity $i$ acquires its new local learning model, $f_i(t+1)$, and determines $e_i(t)$ as follows:

$$e_i(t) = f_i(t+1) - f_i(t). \quad (5.2)$$

4: **Global learning with consensus:** Entity $i$ updates its new estimated global learning model, $F_i(t+1)$, via

$$F_i(t+1) = w_{ii}F_i(t) + \sum_{j \in N_i} w_{ij}F_j(t) + e_i(t), \quad (5.3)$$

where $w_{ii}$ and $w_{ij}$ are constant edge weights defined as follows:

$$w_{ij} = \begin{cases} 
\varphi & \text{if } (i, j) \in E, \\
1 - \varphi d_i & \text{if } i = j, \\
0 & \text{otherwise}.
\end{cases} \quad (5.4)$$

5: **Terminate operation:** Entity $i$ computes the stopping function $S(\Gamma)$ in order to terminate the algorithm or continue.

6: **Output to its neighbours:** Entity $i$ sends its new estimated global learning model, $F_i(t+1)$, to its neighbours.

7: **Next Time:** $t+1$

**Theorem 5.3.1.** Under the Assumptions, in the fixed topology WSANs with the connected communication graph, $\mathcal{G}$, if all $n$ entities follow the Algorithm 4, then:

$$\lim_{t \to \infty} F_i(t) = F^*, i \in \mathcal{N}. \quad (5.5)$$
5.4 Evaluation Results and Analysis

In order to investigate the performance of our proposed algorithm, we designed the simulation evaluation work that 6 sensing entities randomly deployed in the target area with a fixed undirected communication graph as shown in Figure 5.1.

![The fixed undirected communication graph.](image)

5.4.1 Experimental Setup

In the distributed deep RL process, the entities use only local information to make their decisions. The reward function for entity $i$ is designed as:

$$r^i(s^i) = G^i(s^i) - C^i,$$

where $G^i(s^i)$ is a linear function denoting the gain of covered cells in the target area of entity $i$:

$$G^i(s^i) = \text{The number of covered cells} \times \text{gain},$$
and also, $C^i$ corresponds the energy cost for the action of entity $i$:

$$\begin{align*}
C^i = \begin{cases} 
\text{Normal\_Mode\_Cost}, & \text{action} = 2, \\
\text{Low\_Power\_Mode\_Cost}, & \text{action} = 1, \\
\text{Standby\_Mode\_Cost}, & \text{action} = 0.
\end{cases}
\end{align*}$$

(5.8)

Obviously, evaluation works for our proposed algorithm are more persuasive when different environment settings are involved. It can be implemented by varying the parameters, such as gain, Normal\_Mode\_Cost, Low\_Power\_Mode\_Cost, and Standby\_Mode\_Cost, and also working positions. Therefore, different reward functions would be defined, and all the entities should learn different optimal coordination strategies to interact with different environment settings. Table 5.1 shows two environment settings corresponding to two sets of working positions for entities in order to verify our proposed algorithm. Moreover, the deployments of all entities in the target area are described in Fig. 5.2 and Fig. 5.3. In addition, we design a variant CNNs according to the ALexNet which has a compact structure with two convolutional layers ended with two fully connected layers and one output layer. Such a design strategy is intended for maximizing the utilisation efficiency of WSANs resources.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Gains</th>
<th>Normal</th>
<th>Low</th>
<th>Standby</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting 1</td>
<td>0.2</td>
<td>3.3</td>
<td>0.8</td>
<td>0.13</td>
</tr>
<tr>
<td>Setting 2</td>
<td>0.4</td>
<td>4.1</td>
<td>1.0</td>
<td>0.11</td>
</tr>
</tbody>
</table>

As mentioned in the Problem Layout, the input for each entity is the two-dimensional matrix $(9 \times 9)$ with 1 and 0, which is related to the status of blocks around itself: the covered block is 1 and the uncovered block is 0. In fact, we obtain the structure and parameters of CNNs through trial-and-error that it has 3 feature maps with two convolutional layers, two fully connected layers, and one output layer to minimize the cross-entropy cost function and bring about the desired output. Moreover, the first convolutional layer is defined by the shared weights with size = $4 \times 4$, stride = 1 and non-padding as well as a single shared bias, and the second convolutional layer adopt the different size of shared weights, $3 \times 3$. Also, there are 13 neurons in each fully connected layer, and the size of the output layer is 3 related to the number of modes:
Normal Mode, Low-Power Mode, and Standby Mode. In addition, for each neuron, it adopts the ReLU function: \( f(x) = \max(0, x) \), as the non-linear activation function instead of the conventional sigmoid function for the sake of a higher rate of convergence and avoiding the vanishing gradient problem. Also, the \( \Gamma \) for the stopping function, \( S(\Gamma) \), is set to 1000.

### 5.4.2 Results and Analysis

After the parameters of networks are randomly initialized with a robust weight initialisation method [151], each entity takes actions to accumulate its experience reply memory and does not update its model in the first 5000 iterations, next, it performs the proposed algorithm to update its model. Figure 5.4 and Figure 5.5 show the decrease of loss functions for all the entities under different environment settings, and the loss is logged every 2 iterations. Due to no update for all models in the first stage, the loss is thus omitted in the figures. All the entities take the related optimal actions.
after the learning as shown in Figure 5.2 and Figure 5.3.

Generally, the evaluation results illustrate that all the entities manage to learn the optimal policies with the proposed algorithm for different environment settings. Furthermore, it can make great contributions to some cases that collecting the training data in a centralized manner for networks and sharing training data among networks are very expensive and cumbersome, or even impossible as privacy concerns mount and users demand more protections.

5.5 Discussion

In the previous chapters, we presented the shortcomings and drawbacks for traditional outlier clearance approaches and learning frameworks in AIoT systems. Two innovative distributed outlier clearance approaches and a novel distributed learning framework are proposed to solve such problems. Also, we present the analysis of these
proposed methods and the evaluation works in data cleaning and event classifications to demonstrate that our approaches are reliable and effective in the real AIoT applications. Next, we focus on the decision making area in the AIoT systems. In this chapter, a solution is proposed to a class of distributed learning problems in sensing coverage optimisation for WSANs. It allows that each entity in networks establishes the local learning model with the private experience reply memory so as to obtain a
common global learning model via the communication only among its neighbours. It have the advantages in saving the base stations, alleviating networks congestion, etc., compared with the centralised approaches [32,34,80].

And the verification of our proposed approach shows that it is capable of learning the optimal policy. Besides, the proposed method is fully decentralized in the learning phase so that it eliminates the need for transmitting and synchronizing information at the central node compared with centralized methods. Also, the proposed approach performs the learning process and consensus process simultaneously in order to overcome the limitations of traditional distributed learning algorithms. As for future work, it is of interest to study the effect of network topology on the learning performance in WSANs. And also we expect to solve the switching topology and communication time-delay problems in the learning phase.
Chapter 6

Conclusions and Future Works

In this chapter, we would like to present the general conclusions and future works due to the summaries and detailed concluding marks that have been made at the end of every chapter.

6.1 Final Conclusions

This thesis has dealt with the AIoT system with the measurement quality problems and event classification and decision-making efficiency problems. We have finished the full route from initial ideas, via theoretical developments, to the methodologies which can be applied to the related real-life problems. As for the outlier problems, two online distributed methods with low computational complexity and memory usage are proposed. Moreover, the evaluation works on both synthetic and real-life datasets show that our approaches exhibit competitive performance compared with other typical methods. With reference to the distributed learning problems in event classification and decision-making for the AIoT applications, a novel distributed learning framework is presented. The verification of this novel framework on the simulation environment and real-life cases demonstrates that the proposed framework is capable of solving the related distributed learning problems in event classification and decision-making for the AIoT applications. Also, it exhibits the advantages over the centralized method.
and some typical distributed learning frameworks.

In general, we present a series of information fusion approaches for different problems in the AIoT applications. The development and some significant results are reviewed in Chapter 2. Also, the related preliminaries are presented in the same chapter. Next, different outlier clearance approaches have been developed for WSNs (see Chapter 3), and the different applications of the proposed distributed learning framework are described in Chapters 4 and 5, respectively. In the concluding remarks, the contribution of our works and some suggestions for the future research are presented.

To sum up, the main contributions of this thesis can be drawn below.

- To overcome the shortcomings of traditional distributed outlier clearance approaches, a novel distance-based concept is presented to design the outlier factor in networks.

- Different online outlier clearance approaches with remarkable advantages are developed based on the novel outlier factor to solve the related problems in networks. The properties of the novel outlier factor are analysed as well as the evaluation works on the real-life datasets are presented.

- To overcome the shortcomings of traditional learning frameworks introduced in Chapter 2, a novel distributed learning framework with the powerful capability and operational advantages is presented. Moreover, the convergence analysis of it is developed.

- An approach with compelling features solving the distributed deep learning problems in the event classification for WSANs is presented based on our novel distributed learning framework. The evaluation works on the real-life structural health monitoring benchmark datasets indicate that our proposed approach achieves competitive performance.

- A new distributed decision-making approach for the coordinated sensing coverage in networks is presented, which combines our novel distributed learning framework and the deep RL method. Simulation works show that it obtains competitive performance with compelling features.
6.2 Future Works

The following remaining works require further considerations so as to make the argument in this thesis stronger.

1. A good choice of $k$ in KNN algorithms is subject to the datasets. In general, larger values of $k$ reduces effect of the noise on the classification. The investigation of the best choice of $k$ in Chapter 3 need to be established.

2. In the real-life applications, the entities in an IoT system that communicate with each other and need to agree on a specific objective of interest. Therefore, it is possible that some of the existing communication links are able to fail simply due to different reasons. It is important to address agreement problems in their general form for networks with information flow under link failure and creation. Then, the investigation for convergence analysis of our proposed distributed learning framework in a network with switching topology need to be established.

The following open problems could be the heuristic research topics for the future.

1. Incomplete Perception Problems

   In the AIoT applications, due to the limited sensing capabilities of entities in the perception layer or the information loss on account of limited transmission capability in the network layer, the entity in networks might not have a complete perception of the state of the environment. An important challenge in the AIoT applications is making sound decisions with incomplete perception or partially observable states. The traditional model is no longer effective since it might support the wrong decision on optimal action based on the insufficient information of target. The decision can be enhanced with more information to the entity in networks.

2. Input Delay Problems

   Generally, in the AIoT applications such as WASNs applications, we consider that an action is exerted right after the selection is finished by the entity, hence,
the entity would immediately obtain the related reward. Nevertheless, in the real-life AIoT system, there is the control delay for actuators to exert the related actions, and also the feedback from the target environment might be lagging due to different reasons. Such delay or time-lag is always present in the real-life AIoT systems owing to the information transmission, making the decision of the next action, and the transition of the state of the actuator. In particular, input delays exist in an extremely huge AIoT system are too large to be ignored. However, most traditional centralized or distributed algorithms proposed for the AIoT system do not consider these input delays that occurs when sharing information among entities connected to the shared medium. These input delays may be constant, time-varying, or even random variable, degrading the performance of control systems and even impairing the whole AIoT system without considering them.

3. Application of Fog Computing in the AIoT Systems

Due to the traditional information processing architecture of the IoT system is concentrated on the centralized computing or cloud computing, hence, its huge connected entities and massive real-time information transmission would lead to remarkable pressure on the network bandwidth and cloud computing data centres. The cloud computing paradigm can be extended to the edge of networks over fog computing, such that it enables a new breed of applications and services.

4. Synthesis of blockchain technology and AIoT systems

AIoT is leading the contemporary industry to the brand new smart industry featured with data-driven decision-making [159]. However, the intrinsic characteristics of AIoT bring about plenty of challenges, such as the weak privacy protection, complicated interoperability, and security vulnerabilities [160]. Such challenges of AIoT can be solved via the integration of blockchain technology with AIoT. The synthesis of blockchain technology and AIoT systems is named by us as blockchain of autonomous things (BoAT).
Bibliography


[8] Y. He, N. Zhao, and H. Yin, “Integrated networking, caching, and computing


