8-17-2015 12:00 AM

Exploitation of Data Correlation and Performance Enhancement in Wireless Sensor Networks

Tianqi Yu, *The University of Western Ontario*

Supervisor: Xianbin Wang, *The University of Western Ontario*
Joint Supervisor: Abdallah Shami, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

Recommended Citation

Yu, Tianqi, "Exploitation of Data Correlation and Performance Enhancement in Wireless Sensor Networks" (2015). *Electronic Thesis and Dissertation Repository*. 3099.
https://ir.lib.uwo.ca/etd/3099

EXPLOITATION OF DATA CORRELATION AND PERFORMANCE ENHANCEMENT IN WIRELESS SENSOR NETWORKS

(Thesis format: Monograph)

by

Tianqi Yu

Graduate Program in Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Engineering Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

With the combination of wireless communications and embedded system, lots of progress has been made in the area of wireless sensor networks (WSNs). The networks have already been widely deployed, due to their self-organization capacity and low-cost advantage. However, there are still some technical challenges needed to be addressed. In the thesis, three algorithms are proposed in improving network energy efficiency, detecting data fault and reducing data redundancy.

The basic principle behind the proposed algorithms is correlation in the data collected by WSNs. The first sensor scheduling algorithm is based on the spatial correlation between neighbor sensor readings. Given the spatial correlation, sensor nodes are clustered into groups. At each time instance, only one node within each group works as group representative, namely, sensing and transmitting sensor data. Sensor nodes take turns to be group representative. Therefore, the energy consumed by other sensor nodes within the same group can be saved.

Due to the continuous nature of the data to be collected, temporal and spatial correlation of sensor data has been exploited to detect the faulty data. By exploitation of temporal correlation, the normal range of upcoming sensor data can be predicted by the historical observations. Based on spatial correlation, weighted neighbor voting can be used to diagnose whether the value of sensor data is reliable. The status of the sensor data, normal or faulty, is decided by the combination of these two proposed detection procedures.

Similar to the sensor scheduling algorithm, the recursive principal component analysis (R-PCA) based algorithm has been studied to detect faulty data and aggregate redundant data by exploitation of spatial correlation as well. The R-PCA model is used to process the sensor data, with the help of squared prediction error (SPE) score and cumulative percentage formula. When SPE score of a collected datum is distinctly larger than that of normal data, faults can be detected. The data dimension is reduced according to the calculation result of cumulative percentage formula. All the algorithms are simulated in OPNET or MATLAB based on practical and synthetic datasets. Performances of the proposed algorithms are evaluated in each chapter.

Keywords: data correlation, energy efficiency, data fault detection, data aggregation, wireless sensor networks

“Sweat saves blood, blood saves lives,
and brains save both.”
– – Erwin Rommel

Acknowledgments

I would like to express my deepest appreciation to my supervisor, Dr. Xianbin Wang, for his guidance, patience and encouragement in developing my research. It was his enlightening supervisions that inspired me to explore novel research areas and broadened my views in the research area. It was a wonderful and rewarding journey to learn from him.

I also feel grateful to my co-supervisor, Dr. Abdallah Shami. Thanks to his professional insights and technical guidance, I was able to make my rough ideas realized. It was really my honor to work with him.

Sincere thanks to Dr. Raveendra K. Rao, Dr. Jagath Samarabandu and Dr. Sylvia Osborn for being my examination committee. I really appreciate their constructive suggestions on my thesis and research.

I would like to thank my colleague, Dr. Auon Muhammad Akhtar, for his countless and selfless help on my research and my paper submissions. Without the regular meetings and discussions with him, I cannot make such rapid progress in my study.

I feel so lucky that I have an awesome research group. I am the only child of my parents, while the group members are like my brothers and sisters. They helped me a lot in both study and daily life. I would like to extend my thanks to all my friends, especially my roommates. Thanks to them, I feel not lonely in a lonely planet.

Additionally, I also would like to thank every course supervisor and administrative staff that I met in The University of Western Ontario. I cannot complete this degree without their assistance and kindness during these two years.

As always, I feel so grateful to my parents and my family. I really appreciate their love and support throughout this degree and my life. They always back me up no matter how crazy my decisions seem to be.

Contents

List of Figures

List of Tables

List of Appendices

List of Abbreviations

BI	beacon interval
BO	beacon order
CAP	contention access period
CFP	contention free period
CSMA/CA	carrier sense multiple access with collision avoidance
FFD	full function device
GTS	guaranteed time slot
MAC	media access control
NWK	network
PAN	personal area network
PCA	principal component analysis
PHY	physical
PIB	personal area network information base
RFD	reduced function device
RSSI	received signal strength indicator
SD	superframe duration
SO	superframe order
SPCC	sample Pearson correlation coefficient
Wi-Fi	wireless fidelity
WSN	wireless sensor network
ZBR	ZigBee cluster-tree routing

Chapter 1

Introduction

1.1 Overview of Wireless Sensor Networks

With the integrated application of wireless communications technology and embedded systems, wireless sensor networks (WSNs) have attracted significant research attention in recent years. As compared to the traditional wired networks, WSNs have their own specific advantages, *e.g.*, low cost, easy deployment, self-organization and reduced reliance on infrastructure [1]. Due to these advantages, WSNs have already been applied in many different scenarios, especially in some harsh and inaccessible environments. Fig.1.1 summarizes some typical applications of WSNs, including health monitoring, environment monitoring and smart home. Taking smart home as an example, the heating, ventilation and air conditioning systems of a house can be adaptively adjusted by actuators according to sensor readings collected wirelessly. In other words, the applications of WSNs can better serve both public and private daily life of residents at reduced costs. With the increasing popularity in many applications, the number of sensors will be more than 1 trillion by 2025 [2].

A general architecture of WSNs is shown in Fig.1.2, which consists of sink node, data center and a number of wireless sensor nodes [1]. A large number of sensor nodes are randomly deployed in the target areas, in order to monitor and collect the physical parameters of the local environments. Then the sink node is responsible for gathering all the sensor data. Finally, the raw and messy sensor data is analyzed and converted to useful information by the processing and analyses at the data center.

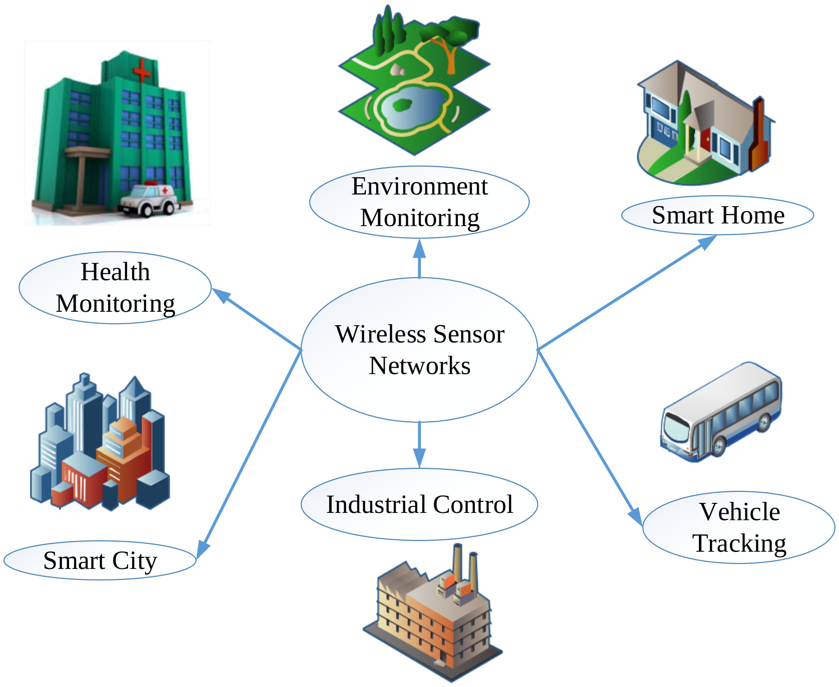


Figure 1.1: Typical applications of wireless sensor networks.

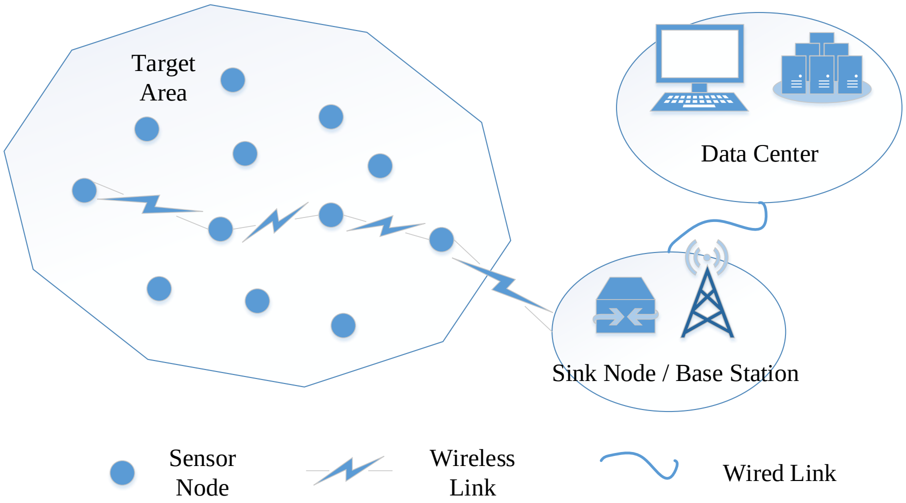


Figure 1.2: General architecture of wireless sensor networks.

However, both the resources and capacities of sensor nodes are quite limited, due to the constraints of low manufacturing and application costs. These inherent limitations can lead to some WSN designing challenges. The first challenge is the energy constraints of sensor nodes and network energy efficiency. Sensor nodes are usually deployed in some inaccessible environments, which makes recharging or replacement of "dead" nodes very difficult. Therefore, how to make full use of the limited energy supply of sensor nodes and improve the network energy efficiency needs to be investigated. Another challenge is the accuracy and efficiency of sensor data. Sensor nodes are usually overly deployed in the target area with substantial redundancy, due to the limited transmission range of sensor nodes and the suboptimal node distribution. The high density of sensor node distribution results in the highly spatial correlation between sensor data, which finally leads to the sensor data redundancy. Additionally, harsh environments make the sensor nodes more vulnerable to different kinds of noise and interference. As a result, the data collected through the sensor nodes may be faulty. The faulty data can lead the data center to a false reaction. Thus, it is also critical to develop techniques in ensuring the quality of sensor data collection.

1.2 Thesis Motivations

Wireless sensor networks have been applied in many different scenarios, due to the multiple advantages discussed earlier. However, there are still some critical challenges needed to be overcome, in order to improve the performance of wireless sensor networks.

Network energy efficiency: Wireless sensor networks are normally used to monitor the harsh and inaccessible environments, in avoiding using the infrastructure based networks. Due to the tough circumstances and the random sensor node deployment, it is really difficult to recharge or replace the "dead" batteries of sensor nodes. Therefore, how to improve the network energy efficiency and prolong the network lifetime is a tough research challenge in WSNs.

Data fault: During the active monitoring period, a large amount of sensor data is generated by the sensor nodes. However, the sensor data collection process may be faulty, due to both the external harsh environments and inner software or hardware malfunctions. Thus, it is necessary to detect and preprocess the faulty sensor data in improving the data accuracy and in avoiding misleading the data center.

Data redundancy: In order to make sure the full sensing coverage, sensor nodes are usually overly deployed in the target areas. The high density of sensor node distribution leads to the highly spatial correlation between sensor readings from neighbor nodes. Data redundancy therefore exists in the correlated sensor data. The transmission and processing of these redundant data can cost extra network resources. Hence, how to aggregate the redundant data poses one additional challenge.

1.3 Research Objectives

The research objectives of this thesis are mitigating the aforementioned problems in wireless sensor networks, namely, network energy efficiency, data fault and data redundancy.

Energy efficiency improvement: The primary research objective of this thesis is improving the network energy efficiency, since the efficiency directly affects the lifetime of WSNs. A general energy consumption model is analyzed first. According to the parameters in the model, data transmission occupies the largest percentage of total network energy consumption. Therefore, the research objective is converted from network energy efficiency improvement to reduction of the data transmissions while ensuring the data reliability at the meantime.

Data fault detection: The faulty sensor data can mislead the data center with erroneous information and lead to possible false reactions at the data center. Since centralized data processing incurs extra burden on the data center and reduces network efficiency, local detection of the faulty sensor data is another research objective of the thesis. By exploitation of the temporal and spatial correlation of sensor data, the status of sensor readings can be diagnosed based on the historical data and the neighbor sensor data. Thus, our specific objective is training a proper time series prediction model and proposing a more accurate neighbor voting algorithm.

Data redundancy reduction: Considering the redundant data consumes extra network resources and reduces the network efficiency, aggregating the redundant sensor readings is also one of the research objectives. A number of mature statistical models have been developed, which could be used to reduce the data dimension and compress the redundant data. Principal component analysis is a conventional data dimension reduction method. Hence, we study the application of the principal component analysis in the research work, in order to reduce the sensor data redundancy.

1.4 Technical Contributions of the Thesis

The main contributions of this thesis are summarized below:

- A spatial correlation based sensor scheduling mechanism is proposed in Chapter 3. Within the proposed technique, a new sensor cluster formation algorithm, termed as adaptive DK-means algorithm, is developed based on the spatial correlation between neighbor sensor data. The new cluster formation algorithm improves the data reliability of the generated clusters. Additionally, a new scheduling algorithm is proposed to decide the order and duration of nodes working as the cluster representative within each cluster. As compared to the baseline ZigBee protocol, the proposed sensor scheduling mechanism reduces the network energy consumption.

- A new temporal and spatial correlation based data fault detection algorithm is proposed in Chapter 4. The variance of physical parameters to be monitored is continuous in nature, so the sensor data from consecutive time instances of the same node and the sensor data from neighboring nodes are highly correlated. Based on the data correlation, a distributed fault detection algorithm is proposed, in order to detect the faulty sensor readings that are not following the normal trends. For temporal correlation based fault detection, three different time series prediction models are compared and then Kalman filter is selected as the prediction model. In terms of the spatial correlation based detection, weighted-median detection is developed. The final detection result is jointly decided by these two detection procedures.

- A recursive principal component analysis (R-PCA) model based data fault detection and data aggregation algorithm is proposed in Chapter 5. The principal component analysis (PCA) model is a commonly used statistical tool for data dimension reduction. In order to enhance the adaptiveness of the model in following the system changes, R-PCA model is proposed based on the modification of conventional PCA model. Squared prediction error (SPE) score and cumulative percentage formula are introduced to assist the R-PCA model implementing the data fault detection and data aggregation operations. Additionally, multivariate sensor readings are considered in the algorithm. In contrast to the conventional algorithms, the proposed multivariate data aggregation algorithm is based on clusters instead of the local nodes so that the network performance can be further improved.

1.5 Thesis Outline

The rest of the thesis is organized as follows:

In Chapter 2, the IEEE 802.15.4/ZigBee protocol stack is briefly introduced first, since it is commonly used in WSNs nowadays. The research in the thesis is also based on this protocol stack. Followed by this, the data correlation in WSNs is numerically analyzed and the related works are summarized according to the different applications. After that, two mathematical models to be used in the thesis are presented. One is the K-means clustering algorithm, the other is the principal component analysis model. These two models are introduced in Chapter 2, as they are the basis of the proposed models in the later chapters.

In Chapter 3, a new sensor scheduling mechanism is proposed so that the network energy efficiency can be improved. In the system model section, the propagation model and energy consumption model are presented. Given the indoor deploy environments for WSNs, the COST-231 indoor propagation model is adopted. Micaz battery model is introduced to evaluate the energy consumption. The proposed sensor scheduling mechanism is then described in detail. Finally, some simulations have been conducted in OPNET to evaluate the novel sensor scheduling mechanism, as compared to the baseline ZigBee protocol.

A new distributed data fault detection algorithm for wireless sensor networks is proposed in Chapter 4. The previous fault detection algorithms are first summarized as related works. Four common types of data faults are introduced as the fault model, which are demonstrated and explained by figures. In the time series prediction model subsection, three different prediction models, *i.e.*, Kalman filter, grey model and auto-regressive moving average model, are introduced and compared. The Kalman filter is used in the proposed algorithm since the residual error and mean absolute percentage error of it are smaller. How the spatial and temporal correlation based distributed fault detection algorithm works is explained through a detailed flowchart. As compared to the previous algorithms in literature, the proposed algorithm improves the detection accuracy proved by the practical and synthetic datasets based simulations.

In Chapter 5, a novel recursive principal component analysis (R-PCA) based data aggregation algorithm is proposed. With the introduction of SPE score, the novel algorithm can detect the data faults at the meantime. In the system model section, cluster tree topology is introduced, since the proposed data aggregation algorithm is cluster based. The R-PCA model is then explained in detail. After that, the R-PCA based multivariate data fault detection and data aggregation algorithm is proposed. Finally, simulation results show that the novel algorithm improves the data fault detection accuracy, improves the data restoration accuracy and reduces the network energy consumption.

Lastly, all the contributions presented in the previous chapters are concluded in Chapter 6. The plan for the future research is discussed in this Chapter as well.

Chapter 2

Data Correlation Analysis and Modelling in Wireless Sensor Networks

2.1 Overview of IEEE 802.15.4/ZigBee

Bluetooth [3] and ZigBee [4] protocols were both designed for personal area networks (PANs) in the specification of IEEE 802.15. However, most WSNs adopt ZigBee protocol instead of Bluetooth for the following reasons:

Energy Consumption: Bluetooth consumes more power than ZigBee. ZigBee devices are about 2.5~3 times more energy efficient than Bluetooth devices under the same conditions.

Self Organization: ZigBee networks support the self-organization and self-healing technology. In contrast, Bluetooth does not have this capacity.

Scalability: Bluetooth works in a master-slave mode and the master node can support only up to 7 slaves. By contrast, the topologies of ZigBee networks are more flexible and have different variations, namely, star, mesh and cluster topologies. The flexible topologies make the ZigBee networks more scalable. The number of connected nodes can be up to 65,000.

Therefore, ZigBee protocol is widely used in WSNs for its low energy cost, self-organization capacity and better scalability. This subsection briefly introduces the ZigBee protocol stack. The protocol stack architecture is shown in Fig.2.1 [4]. It can be seen that the PHY layer and MAC layer protocols in the stack are defined by the IEEE 802.15.4 working group [5], while the specifications of the upper layers are designed by the ZigBee alliance [4].

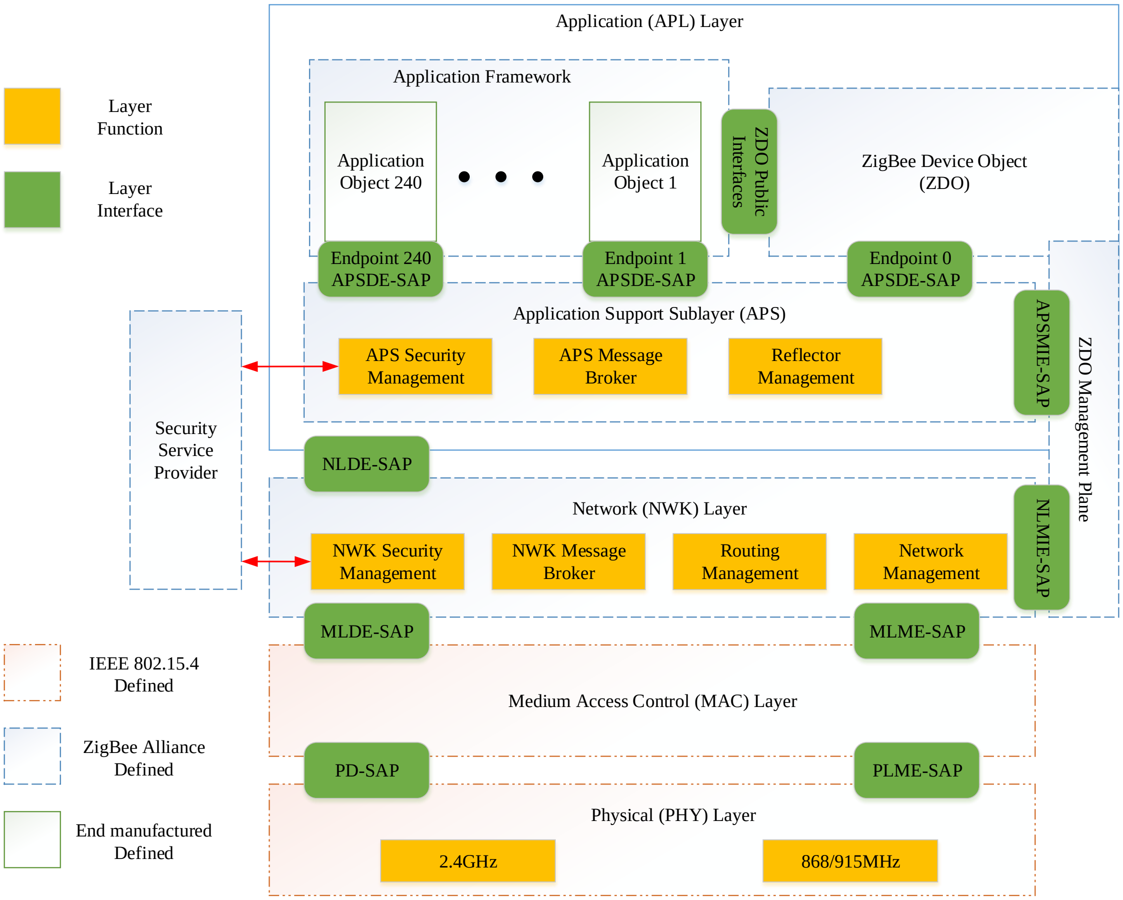


Figure 2.1: Architecture of ZigBee protocol stack.

2.1.1 IEEE 802.15.4 PHY/MAC Protocols

The IEEE 802.15.4 standard supports three different working frequency bands at the PHY layer, *i.e.*, 868MHz, 915 MHz and 2.4 GHz. Data rate at 2.4 GHz is 250 kbps, which is higher than 40 kbps at 915 MHz and 20 kbps at 868 MHz. The working channel can be chosen dynamically. The specific working channels at different frequency bands are shown in Fig.2.2. Additionally, the input signal power thresholds at receiver end are -85 dBm at 2.4 GHz and -92 dBm at 868/915 MHz, respectively.

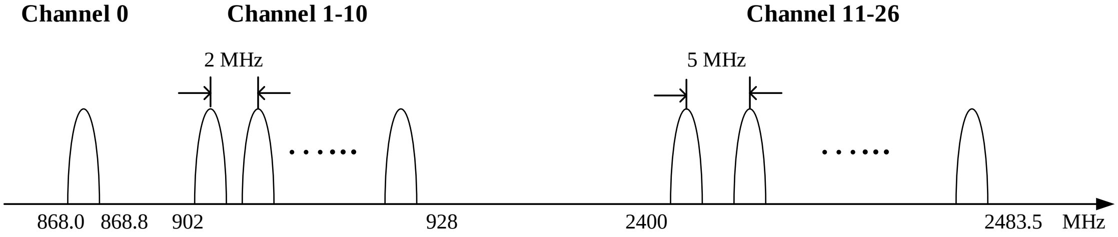


Figure 2.2: IEEE 802.15.4 physical layer channels.

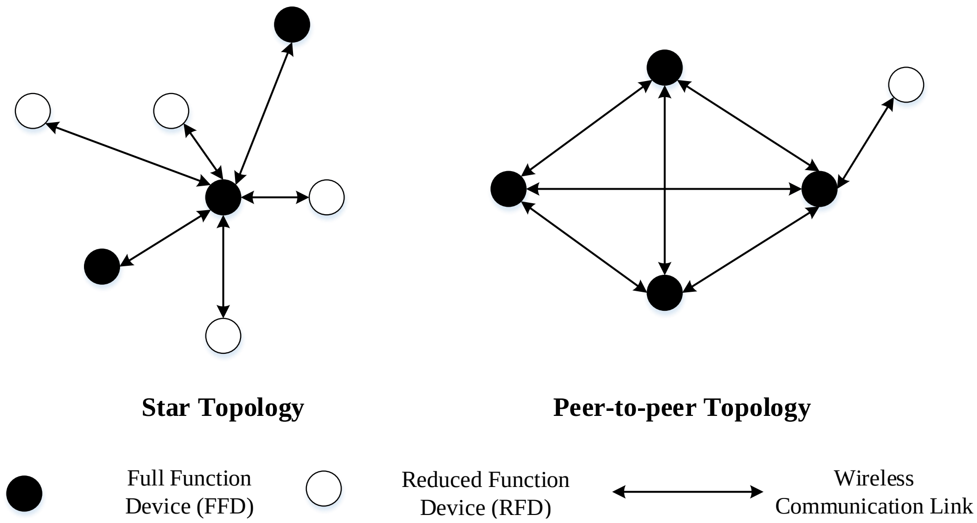

Star Topology
Peer-to-peer Topology
Full Function Device (FFD)
Reduced Function Device (RFD)
Wireless Communication Link

Figure 2.3: Examples of star topology and peer-to-peer topology in IEEE 802.15.4.

There are two types of devices in an IEEE 802.15.4 network, *i.e.*, full function device (FFD) and reduced function device (RFD). The difference between FFD and RFD is that the RFD does not have the routing capacity. Hence, an FFD can perform all the roles in the network, a personal area network (PAN) coordinator, a coordinator or an end-device, while RFD can only be an end-device and communicate with an FFD. Two basic topologies that can be set up by the devices in IEEE 802.15.4 are star topology and peer-to-peer topology. Examples of the two basic topologies are shown in Fig.2.3 [5].

IEEE 802.15.4 offers two different working modes at its MAC layer, namely, nonbeacon-enabled mode and beacon-enabled mode. Data frames in nonbeacon-enabled mode compete to occupy the communication medium by unslotted CSMA/CA scheme. In beacon-enabled mode, IEEE 802.15.4 defines a new frame type, termed as superframe. The superframe consists of active and inactive sections. This structure better satisfies the low energy consumption requirement of the network, since the devices only interact during the active section and enter a low-power (sleep) mode during the inactive section. The active section is further divided into 16 slots. Beacon signal is sent by the coordinator at the first time slot of the superframe, which defines the duty-cycle of the devices and ensures the synchronization of the network. The length of superframe is defined by the beacon interval (*BI*) between two beacon signals as

$$BI = aBaseSuperframeDuration \times 2^{BO}\ symbols, \tag{2.1}$$

where $aBaseSuperframeDuration$ refers to the number of symbols forming a superframe and BO is the beacon order, an attribute in MAC PIB (personal area network information base) used to specify how often the beacon signal is sent, $0 \leq BO \leq 14$. The active section of BI is termed as superframe duration (SD), given by

$$SD = aBaseSuperframeDuration \times 2^{SO}\ symbols, \tag{2.2}$$

where SO is the superframe order, an attribute in MAC PIB used to specify the length of the active section of the superframe, $0 \leq SO \leq BO$. Finally, the duty cycle is defined as the ratio of SD to BI, *i.e.*, *Duty-cycle* $= 2^{SO-BO}$.

Additionally, the active section of superframe is separated into two phases, i.e. contention access period (CAP) and contention free period (CFP). In CAP, data frames compete to transmit by exploitation of slotted CSMA/CA scheme, while data is transmitted without competition in the previously assigned guaranteed time slots (GTS) of CFP. The specific structure of superframe is shown in Fig.2.4 [5].

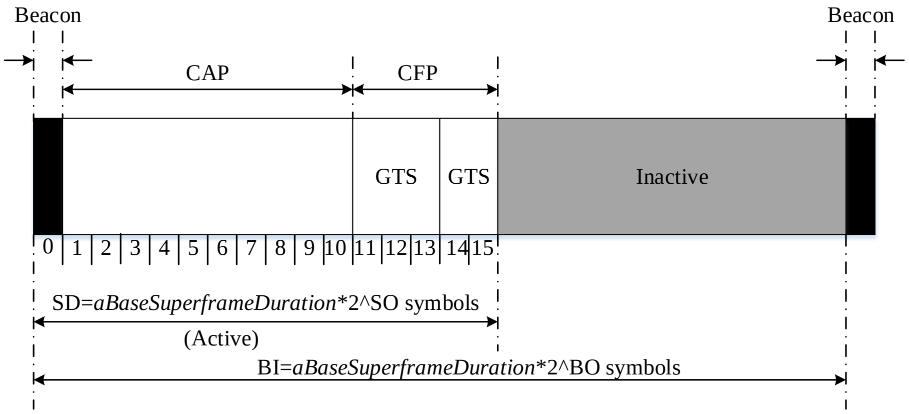

Beacon
Beacon
CAP
CFP
GTS
GTS
Inactive
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
SD=*aBaseSuperframeDuration**2^SO symbols
(Active)
BI=*aBaseSuperframeDuration**2^BO symbols

Figure 2.4: Superframe structure in IEEE 802.15.4.

2.1.2 ZigBee Cluster-tree Routing Protocol

ZigBee cluster-tree routing (ZBR) protocol cooperates with IEEE 802.15.4 PHY and MAC layer protocols, which is designed specifically for the resource constrained sensor networks. The multi-hop routing of ZBR is implemented by the distributed addressing assignment mechanism [4]. With this mechanism, every device within the network gets a unique address assigned by the PAN coordinator.

Every coordinator/router in a ZigBee network is capable to support $nwkMaxChildren$ (C_m) child nodes at most (including $nwkMaxRouters$ (R_m) routers). Every device is assigned an associated depth d, which refers to the minimum number of hops to the PAN coordinator through only parent-child links. Moreover, the maximum depth of a ZigBee network is given by $nwkMaxDepth$ (L_m) [4]. Given a node with address A_{parent}, the address of its m^{th} router child is given by

$$A_m = A_{parent} + C_{skip}(d) \cdot (m - 1) + 1, \tag{2.3}$$

where $1 \leq m \leq R_m$. While the address of its n^{th} end device child is given by

$$A_n = A_{parent} + C_{skip}(d) \cdot R_m + n, \tag{2.4}$$

where $1 \leq n \leq (C_m - R_m)$, and

$$C_{skip}(d) = \begin{cases} 1 + C_m \cdot (L_m - d - 1) & , \ if \ R_m = 1, \\ \dfrac{1 + C_m - R_m - C_m \cdot R_m^{L_m - d - 1}}{1 - R_m}, & otherwise. \end{cases} \tag{2.5}$$

For example, the parameters of the network are $nwkMaxChildren(C_m) = 5$, $nwkMaxRouters(R_m) = 3$ and $nwkMaxDepth = 3$. First, $C_{skip}(d),\ d = 0 \sim 3$ is calculated and listed in Table 2.1. Then the address allocation is demonstrated in Fig.2.5.

Table 2.1: C_{skip} Values of Different Depths

depth (d)	$C_{skip}(d)$
0	21
1	6
2	1
3	0

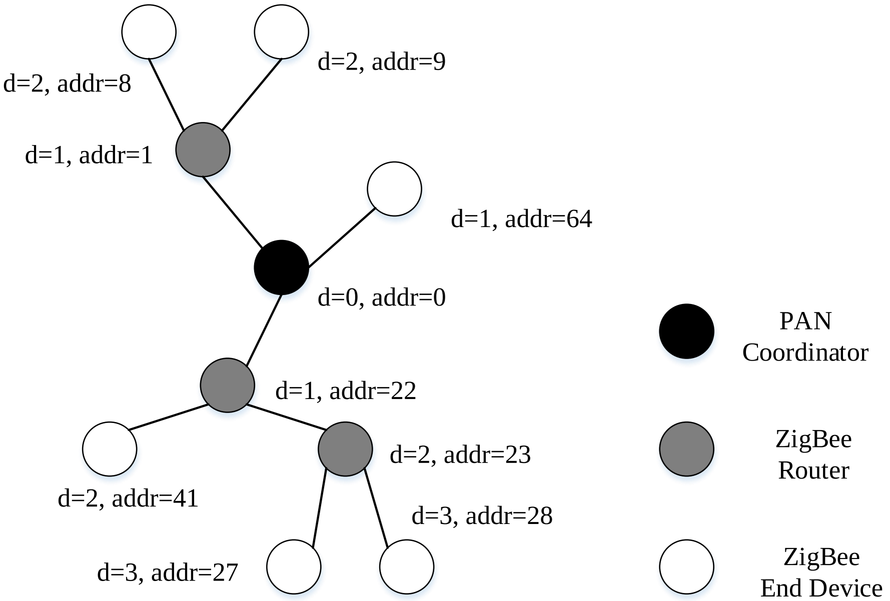


Figure 2.5: Example of address allocation.

Given the unique ZigBee addresses, routing procedure could be greatly simplified. Suppose a routing destination with address D, a router with address A decides whether D is its descendant. If (2.6) is satisfied, router A identifies D as its descendant and sends data packet to one of its children. Otherwise, router A sends data packet back to its parent.

$$A < D < A + C_{skip}(d-1). \tag{2.6}$$

2.2 Data Correlation in WSNs

Our research in this thesis is based on the intrinsic correlation between sensor readings. There are some recent works that focus on the temporal and spatial correlation of sensor data in literature. In this section, the data correlation in WSNs is investigated based on both numerical analysis and literature survey.

2.2.1 Data Correlation Analysis

Sample Pearson correlation coefficient (SPCC) [6] is used to evaluate the extent of data correlation, which is calculated as

$$r_{x,y} = \frac{\Sigma_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\Sigma_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\Sigma_{i=1}^{n}(y_i - \overline{y})^2}}, \tag{2.7}$$

where x, y are two vectors, $\overline{x}$ and $\overline{y}$ are the mean values of the vectors and n is the number of samples in the vector. The value of $r_{x,y}$ falls in $[-1, 1]$. If $r_{x,y} = 0$, it means that x and y are uncorrelated. If $r_{x,y} = -1$ or $r_{x,y} = 1$, x and y are absolutely linearly correlated.

Proof of spatial correlation In order to prove the spatial correlation between sensor readings, the SPCC values of temperature readings from Node 31 ~ 40 in Intel Berkeley research lab (Fig.2.6) are calculated [7]. More specifically, the first two hundred temperature samples on 2/28/2004 are used. The SPCC values are listed in Table 2.2.

Table 2.2: Sample Pearson Correlation Coefficient

	Node 31	Node 32	Node 33	Node 34	Node 35
Node 31	1	0.9934	0.9886	0.9927	0.9956

	Node 36	Node 37	Node 38	Node 39	Node 40
Node 31	0.9924	0.9846	0.9744	0.9900	0.9839

It can be seen that the SPCC values in Table 2.2 are all over 0.97, while the correlation test

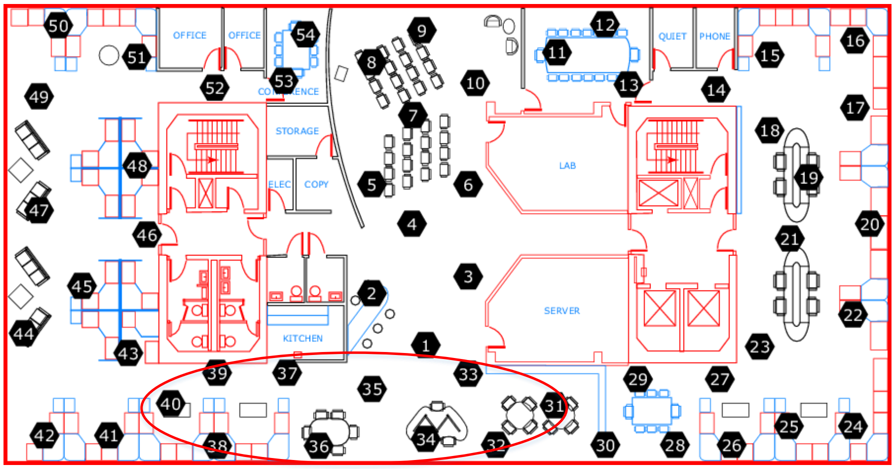


Figure 2.6: Deployment of Node 31 ~ 40 in Intel Berkeley research lab.

threshold in t-test at 0.05 α-level is 0.171 (sample size=200) [8]. Therefore, it indicates that the temperature readings from neighbor nodes are highly correlated.

Proof of temporal correlation In order to prove the temporal correlation of sensor readings, SPCC of the continuous temperature readings is calculated and demonstrated in Fig.2.7, where

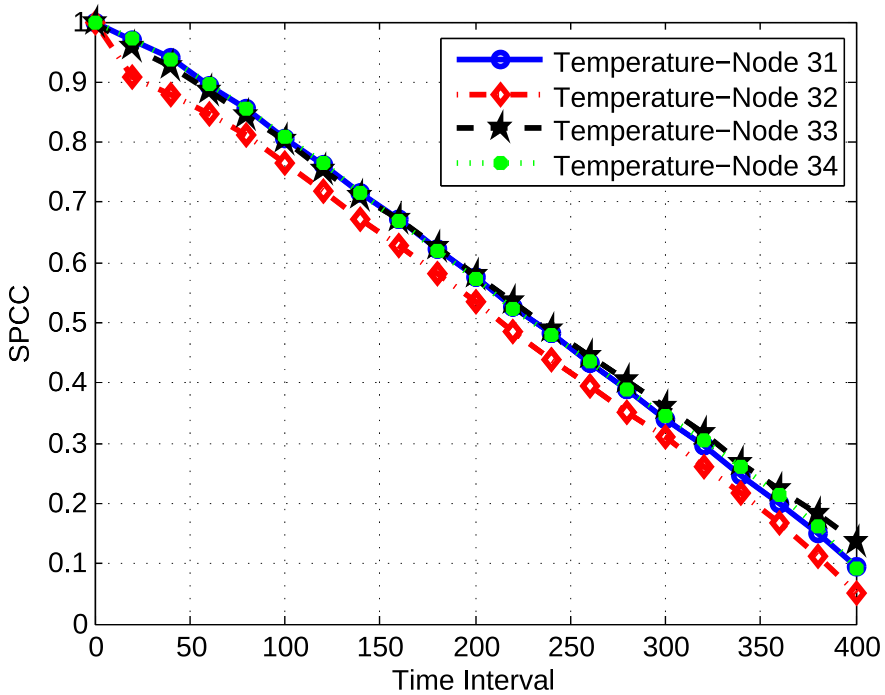


Figure 2.7: SPCC of temperature readings with different time intervals.

time interval refers to the number of time instances between the two sampling vectors and each time instance refers to the sampling gap, *i.e.*, 30s.

Fig.2.7 shows that the SPCC values of temperature readings decrease with the increment in time interval. The decrement indicates that the strength of the correlation between temperature readings in temporal domain is reducing due to the increased time interval. Additionally, the SPCC is below the correlation test threshold when the time interval is larger than 350 time instances. This means that the correlation between temperature readings does not exist after a certain time interval. Therefore, it is needed to be careful with the selection of historical observations when exploit the temporal correlation.

2.2.2 Data correlation in WSNs and Related Applications

In recent years, a number of researchers have worked on sensor data correlation in WSNs and used in different applications. Some related works have been summarized here and categorized by different aims.

2.2.2.1 Data Correlation based Energy Efficiency Improvement

An energy-efficient data collection framework is proposed based on both temporal and spatial correlation of sensor data [9]. The framework is realized by the cluster based localized prediction algorithm, where the clusters are formed up according to the extent of spatial correlation. Specifically, the predicted data value is used at the cluster head instead of practical data generated by the sensor node so that the energy consumed by the real-time data transmission can be saved. The database at cluster head is updated only when the difference between predicted data and practical data is over a certain threshold. Furthermore, the framework can be accommodated with sleep/awake scheduling and data aggregation as well.

Similarly, *L.Xiang et al.* [10] focus on the problem of energy-efficient data collection in WSNs as well. In their work, a novel compressive data aggregation (CDA) scheme based on the compressive sensing (CS) model is proposed. At the receiver end, diffusion wavelet is used to recover the original data so that the recovery accuracy can be ensured regardless of the network topology. In terms of the routing cost, the authors first prove that the minimum-energy compressive data aggregation is an NP-complete problem, and then both optimal and heuristic solutions are proposed. The experiments are conducted based on both practical and synthetic databases. Results show that the proposed CDA scheme improves both the data recovery accuracy and the network energy efficiency.

A distributed adaptive sparse sensing (DASS) algorithm is proposed based on compressive sensing and incremental PCA models [11]. Different from other works, DASS considers not only the energy consumed by data transmission, but also the sensing energy cost. DASS estimates where and when to sense the physical field based on spatial-temporal correlation, in order to reduce the sensing energy consumption. The experimental results indicate that the consideration of sensing costs can further reduce the network energy consumption indeed.

2.2.2.2 Data Correlation based Fault Detection

Some works aiming at data fault (or data outlier) detection in WSNs exploit the sensor data correlation as well, [12][13][14]. An exceptional message supervision mechanism (EMSM) is proposed by exploitation of the spatial-temporal correlation [12]. In EMSM, cosine similarity is used as the metric to evaluate the extent of correlation and detect the fake messages. With the introduction of EMSM, the security attacks in routing protocols can be detected and processed.

Works on anomaly detection within a non-stationary environment have been summarized as a survey [13]. The authors analyze and demonstrate different categories of data anomalies in the literature first, [15][16][17]. Then the characteristics of data distribution in non-stationary environment are discussed. Based on these concepts, multiple data anomaly detection algorithms are introduced and categorized according to different change detection, model selection, sliding window and model construction methods at last.

Different from previous works, a novel segment-based anomaly detection algorithm is proposed in order to detect the long-term data anomalies [14]. The novelty of the algorithm is that it is segment-based instead of point-based. A novel detection metric based on the covariance matrices of neighbor sensor data is presented, termed as prediction variance. The detection threshold relies on the centralized analysis of sample covariance matrices at cluster head. In order to reduce the cost of centralized analyses, the Spearman's rank correlation coefficient and differential compression are used to compress the raw covariance matrices. Receiver operating characteristic curves in the simulation results show that the proposed algorithm performs better in long-term data anomaly detection while weaker in the random outlier detection.

2.2.2.3 Data Correlation based Data Aggregation

Except for the data fault detection, temporal and spatial correlation can be used to aggregate the redundant sensor data as well. One of the major research objectives of the data aggregation algorithms is reducing the restoration error (*i.e.*, improving the recovery accuracy) of the aggregated data at the receiver end. Some related works are presented as follows.

An iterative algorithm for compressing matrices calculation has been proposed, in order to minimize the restoration error at fusion center [18]. The fusion center is responsible for collecting and recovering the reduced-dimensional data matrices from leaf sensor nodes. Applications of the iterative algorithm at two different noise scenarios are discussed, namely, homogeneous and inhomogeneous environments. Simulations at two scenarios are conducted, respectively. Simulation results show that the restoration error at fusion center is tolerant.

A novel distributed data storage (DDS) coding scheme is proposed based on both spatial and temporal correlation of sensor data, termed as spatial-temporal compressive network coding (ST-CNC) [19]. More specifically, two dimensional compressive sensing model is used to encode and transmit sensor data in ST-CNC. For data recovery, Kronecker structure framework is exploited to ensure the small number of data receptions and high data recovery accuracy. Simulation results prove that the proposed scheme reduces the number of data transmissions without the sacrifice of data recovery accuracy.

A novel data aggregation algorithm in WSNs is proposed based on characteristic correlation approach [20]. The basic principle behind characteristic correlation approach is still the spatial correlation between neighbor sensor data. The difference is that they upgrades the criteria of data correlation to the identical data magnitude and data gradient. They set up both the practical and simulated experiments, in order to evaluate the proposed aggregation algorithm. The experimental results show that the proposed algorithm improves the data restoration accuracy because of the more critical correlation criteria.

2.3 K-means Clustering Algorithm

As mentioned in Section 2.1, cluster tree topology is commonly used in wireless sensor networks. Therefore, cluster formation algorithms for WSNs have attracted much attention from researchers recently. With the help of the algorithms, correlated elements can be clustered by one or more metrics, *e.g.*, Euclidean distance. K-means clustering algorithm is a classical cluster formation algorithm in machine-learning, which was first proposed in [21]. Elements are clustered into K groups according to the minimal Euclidean distance principle, where the number of groups (K) and the initial group centroids are set at the beginning. In recent years, the K-means algorithm has been introduced into wireless sensor networks to cluster the sensor nodes. In this section, the algorithm and its related applications are discussed.

2.3.1 Algorithm Outline

The K-means clustering algorithm works as follows.

- **Step 1:** Select K initial cluster centroids.
- **Step 2:** Assign element i ($i = 1, \ldots, M$) to cluster j with minimum $d(i, C(j))$. Then, $IC(i) = j$ and increment $NC(j)$ by 1.
- **Step 3:** Update the cluster centroid of cluster j ($C(j)$) with the average value of all the cluster members.
- **Step 4:** Repeat step 2,3 until all the elements are clustered.

Abbreviations used in the algorithm description are summarized in Table 2.3.

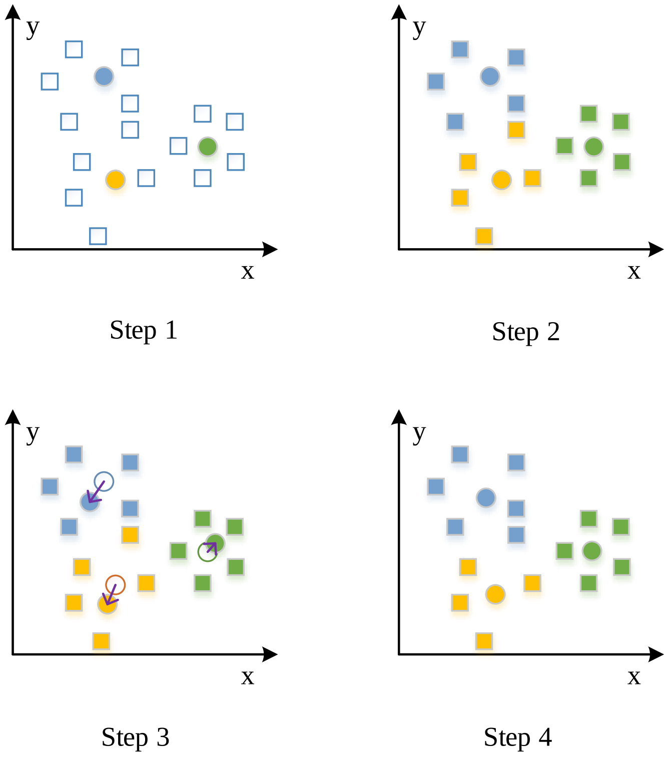
Clustering procedures are described in Fig.2.8, where the squares refer to the basic elements while the circles stand for the cluster centroids.

2.3.2 K-means clustering algorithm in WSNs

As a classical cluster formation algorithm, K-means clustering algorithm has already been widely used in WSNs. Recent works in literature are summarized as follows. A novel hybrid clustering formation algorithm (QK-means) is proposed in [22], which is based on both

Table 2.3: Abbreviations in K-means clustering algorithm

K: number of clusters
M: number of elements in total
N: dimension of element
j: cluster
$C(j)$: cluster centroid of j
$d(i, C(j))$: Euclidean distance between element i and cluster centroid of j
$NC(j)$: number of elements assigned to cluster j
$IC(i)$: the cluster that contains element i



Step 1

Step 2

Step 3

Step 4

Figure 2.8: Procedures of K-means clustering algorithm.

conventional K-means algorithm and the community detection in complex networks. Community detection refers to the clustering process in a complex network, where a complex network means the large scale sensor network with non-trivial topology. After the complex network is separated into several large clusters by community detection, the conventional K-means algorithm is implemented to further part the large clusters into small ones. Simulation results prove that the proposed QK-means algorithm improves the coverage rate and decreases the number of lost messages.

Different from the univariate model [22], *F.Medhat et al.* [23] consider the multivariate data model. Based on the multivariate model, three different clustering algorithms are compared, *i.e.*, fuzzy c-means algorithm, k-means algorithm and LEACH-C algorithm. According to the simulation results, fuzzy c-means algorithm and k-means algorithm outperform the conventional clustering algorithm in WSNs (LEACH-C) on prolonging the network lifetime.

Due to the weakness of LEACH-C algorithm [23], *D.Mechta et al.* [24] improve the conventional LEACH-C algorithm and propose a new routing protocol based on K-means algorithm and minimum transmission energy routing protocol, termed as LEACH-CKM. As compared to the the baseline LEACH-C algorithm, LEACH-CKM adopts k-means algorithm as the formation method so that the node isolation and information transmission failure problems can be solved. Simulation results demonstrate that the change in LEACH-CKM improves the amount of data received at base station and prolongs the network lifetime by 30%.

A delay-aware data collection network structure based on k-means algorithm (DADCNS-RK) is proposed, so that the total propagation distance can be reduced and the energy consumed by data transmission can be decreased by recursively minimizing the sum of Euclidean distances between sensor nodes [25].

The k-means algorithm is integrated with the prefix frequency filtering (PFF), in order to reduce the latency caused by the high complexity of basic PFF [26]. Specifically, the sensor nodes are clustered by the k-means algorithm first. Then prefix frequency filtering is performed within the clusters instead of global range. As compared to the global PFF, cluster based PFF aggregates the redundant data without sacrificing too much time complexity.

2.4 Principal Component Analysis

2.4.1 Introduction to Principal Component Analysis

Principal component analysis (PCA) is a statistical tool which is always used to reduce the dimension of data [27]. The basic mathematical principle behind PCA is the change of basis. Given a matrix, its natural basis is an identity matrix I, *i.e.*, $X = IX$, where

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}_{m\times m}, \tag{2.8}$$

and X is the original data matrix consisted of m physical measurements and n observations.

PCA aims at finding out a basis P, which projects X (an $m \times n$ matrix) into a linearly uncorrelated matrix Y with reduced dimension, *i.e.*, $Y = PX$. Since Y is linearly uncorrelated, its covariance matrix is diagonal. Mathematically,

$$C_Y = \frac{1}{n-1} YY^T = \Lambda, \tag{2.9}$$

where Λ is a diagonal matrix. Specially, the concept of covariance matrix is defined in [27].

Therefore, the problem of determining the proper basis P is simplified as decomposing the original data matrix X so as to diagonalize the covariance matrix of Y.

2.4.1.1 Decomposition Method

Two commonly used decomposition methods in principal component analysis are introduced in this subsection. The two decomposition methods are based on the eigenvectors and singular values of the covariance matrix of X (*i.e.*, $\frac{1}{n-1}XX^T$), respectively.

Eigenvector Decomposition Method This decomposition method is based on the theorem that given a symmetric matrix S, it can be diagonalized by its orthogonal eigenvector matrix.

Therefore, the covariance matrix of X can be diagonalized by its eigenvector matrix. It is mathematically presented as,

$$C_X = \frac{1}{n-1} XX^T = E\Lambda E^T, \tag{2.10}$$

where the columns in E are the eigenvectors of C_X and Λ is a diagonal matrix with the eigenvalues of C_X. Set P equal to E^T. Based on (2.10), it can be further derived that

$$\begin{aligned} C_Y &= \frac{1}{n-1} YY^T \\ &= P\frac{1}{n-1} XX^T P^T \\ &= PC_X P^T \\ &= PP^T \Lambda PP^T \\ &= \Lambda. \end{aligned}$$

It can be seen that C_Y is diagonal when P is set to E^T. Now, the transformation basis P is set to the transposition of eigenvector matrix of C_X and the j^{th} diagonal value of C_Y is the variance of original data matrix X along P_j.

Singular Value Decomposition Method Given an $n \times m$ data matrix Z ($Z \equiv \frac{1}{\sqrt{n-1}} X^T$), $Z^T Z$ is an $m \times m$ square and symmetric matrix with rank r. Based on (2.11),

$$(Z^T Z)\hat{v}_i = \lambda \hat{v}_i, \tag{2.11}$$

the eigenvectors of $Z^T Z$ (*i.e.*, $\{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_r\}$) and corresponding eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_r\}$ can be calculated. Furthermore, the singular value is defined as

$$\sigma_i \equiv \sqrt{\lambda_i}. \tag{2.12}$$

The set of $n \times 1$ vectors $\{\hat{u}_1, \hat{u}_2, \dots, \hat{u}_r\}$ is given by

$$\hat{u}_i \equiv \frac{1}{\sigma_i} Z\hat{v}_i. \tag{2.13}$$

It can be further derived that

$$\hat{u}_i \cdot \hat{u}_j = \begin{cases} 1, & if\ i = j, \\ 0, & otherwise. \end{cases} \tag{2.14}$$

Additionally, $||Z\hat{v}_i|| = \sigma_i$. Therefore, the scalar version of singular value decomposition is just a restatement of (2.13),

$$Z\hat{v}_i = \sigma_i \hat{u}_i. \tag{2.15}$$

We set Σ as the diagonal matrix consisted of singular values, U and V stand for the sets of $\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_r\}$ and $\{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_r\}$ appended with $(n - r)$ and $(m - r)$ zeros, separately. Then the finalized decomposition equation is

$$Z = U\Sigma V^T, \tag{2.16}$$

which indicates that the matrix Z can be decomposed into the multiplication of two orthogonal matrices and one diagonal matrix.

After the decomposition procedure, the matrix Y is linearly uncorrelated. But the dimension of Y is still the same as X. Hence, how to extract the principal components of X and make Y less dimensional is further investigated.

2.4.1.2 Number of Principal Components

Given the eigenvalue matrix Λ, the eigenvalues λ_i in Λ are reordered by their values. The eigenvalue with new order is labelled as $\tilde{\lambda}_i$. The largest l eigenvalues in the matrix are extracted as principal components, where the value of l is given by

$$\alpha = \frac{\sum_i^l \tilde{\lambda}_i}{\sum_i^m \tilde{\lambda}_i} \times 100\%, \tag{2.17}$$

where α is determined by the specific application requirements, *e.g.*, 90%.

The dimension of eigenvalue matrix is reduced from m to l, while the associate eigenvector matrix is reordered and reduced at the meantime, $\tilde{E}_l$. Meanwhile, the transformation basis P is updated to $\tilde{E}_l^T$. Finally, the generated matrix Y is a less dimensional and linearly uncorrelated matrix, namely, $Y = PX = \tilde{E}_l^T X$. The method (2.17) is termed as cumulative percentage formula. There are also other methods that can be used to determine the value of l in literature, *e.g.*, Kaiser's rule.

2.4.1.3 PCA based Data Fault Detection Metric

Principal component analysis itself can only be used to reduce the data dimension. Therefore, some auxiliary metrics are needed to detect the faults in sensor readings. Two commonly used metrics are briefly introduced as follows [28].

SPE Score SPE score for matrix $\overline{x}$ is calculated as

$$SPE(t) = \|\overline{x}(t) - \widehat{x}(t)\|_2^2, \tag{2.18}$$

where $\overline{x}(t)$ is the off-mean matrix of the original $x(t)$ and $\widehat{x}(t)$ is the approximate matrix calculated as

$$\widehat{x}(t) = \tilde{E}_l(t)\tilde{E}_l^T(t)\overline{x}(t), \tag{2.19}$$

$\tilde{E}_l(t)$ is the reordered and reduced-dimensional version of E, where E is the eigenvector matrix of $C_{\overline{X}}$ ($\overline{X}$ is the off-mean sensor data).

The threshold of SPE score is defined as

$$\Gamma_{SPE}(t) = \theta_1(t)\left[\frac{h_0\xi\sqrt{2\theta_2(t)}}{\theta_1(t)} + 1 + \frac{\theta_2(t)h_0(t)(h_0(t)-1)}{\theta_1(t)^2}\right]^{\frac{1}{h_0(t)}}, \tag{2.20}$$

where

$$h_0(t) = 1 - \frac{2\theta_1(t)\theta_3(t)}{3\theta_2^2(t)}, \tag{2.21}$$

$$\theta_j(t) = \sum_{i=l+1}^{m} N\tilde{\lambda}_i^j(t), \tag{2.22}$$

$$\xi : P\{-\xi < X < \xi\} = 0.99, \tag{2.23}$$

where X follows normal distribution.

T^2 **Score** The definition of T^2 score for matrix $\overline{x}$ is given by

$$T^2(t) = \|\overline{x}^T(t)\tilde{E}_l(t)\tilde{\Lambda}_l^{-1}(t)\tilde{E}_l^T(t)\overline{x}(t)\|_2^2, \tag{2.24}$$

where $\tilde{\Lambda}_l(t)$ contains the first l largest eigenvalues of $C_{\overline{X}}$.

The threshold of T^2 is calculated as

$$\Gamma_{T^2}(t) = \frac{(m^2-1)l}{m(m-l)}\xi_F, \tag{2.25}$$

where

$$\xi_F = F^{-1}(P\{-\xi < X < \xi\}), \tag{2.26}$$

where F^{-1} refers to the inverse F distribution.

2.4.2 Principal Component Analysis in WSNs

Principal component analysis is always used to aggregate the large amount of messy and redundant sensor data from wireless sensor networks. Some recent works focused on PCA based data aggregation in WSNs have been summarized here.

LocalPCA algorithm is a distributed data aggregation algorithm based on the principal component analysis [29]. In their work, tree topology is used to set up the network and the sensor data is transmitted from the leaf sensor nodes to the sink node along the tree. At each level of the tree, the sensor data from lower levels is aggregated first before being forwarded. Simulation results show that the LocalPCA algorithm reduces the reconstruction error, as compared to the PCAg and EAPCAg algorithms.

Multivariate sensor data sampling algorithm considers the multiple variables in WSNs [30]. The algorithm consists of three steps. First is component transformation. PCA, robust PCA and ICA are all introduced and compared. Second step is ranking, which sorts the first component values. The third step is sampling. Both variance (ANOVA) and maximum relative absolute error are used to evaluate the reconstruction error. However, different from other PCA based aggregation algorithms, the proposed sampling algorithm extracts the principal components along the time series.

Schemes iPC3 and oPC3 are proposed based on the PC3 model (principal components based context compression model [31]), in order to implement multivariate data aggregation in WSNs [32]. iPC3 exploits incremental principal component analysis (IPCA) to implement the online learning of the eigenbasis (*i.e.*, basis of PCs), thus the learning phase of PC3 (*i.e.*, the phase during which the eigenbasis is calculated) is eliminated. oPC3 uses optimal stopping theory (OST) to decide the best time to switch from aggregation to learning phase. Relative reconstruction error is used to evaluate the algorithm performance. The simulation results show that iPC3 and oPC3 definitely improve the data reconstruction accuracy.

Additional to data aggregation, principal component analysis can be also used to detect data fault (or data anomaly, data outlier) in WSNs with the help of SPE score or T^2 score.

A PCA-based data fault detection algorithm has been proposed in [33]. In the conventional PCA model, the off-mean method is used to preprocess the raw data, while the raw sensor data is standardized to zero-mean and unit-variance data set in the proposed work so that the influence of different scales can be reduced. The subspace-based anomaly detection algorithm is used to diagnose the data outliers. More specifically, the number of principal components (PCs) is determined by the Kaiser's rule and the squared prediction error (SPE) score is adopted as the detection criterion. Simulation results show that the proposed algorithm can be used to detect both random and correlated data anomalies.

Two approaches have been proposed to detect the data anomalies in wireless sensor networks in [34]. The first approach is based on a multivariate Gaussian model, which exploits the intrinsic correlation of sensor data. In the second approach, the authors present the kernel PCA, where the geometric information is included as well. Different from the general PCA, the kernel PCA eliminates the limitation of data linearity. *S.Chan et al.* [28] preview the conventional PCA model and two auxiliary fault detection metrics first, and then new robust recursive fault detection algorithm is proposed based on the previous work [33]. The new robust recursive model and fault detection algorithm aim at being sensitive to the gradual system changes and robust to the dramatical data outliers. Both simulated and practical experiments have been conducted in [28]. The experimental results show that the proposed robust recursive algorithm outperforms the conventional algorithms on detection accuracy and false alarm ratio .

2.5 Chapter Summary

In this chapter, some basic concepts and models used in the thesis are reviewed. Since the IEEE 802.15.4/ZigBee protocol stack is used in most of the wireless sensor networks, brief introduction to the protocol stack is given at the beginning of this chapter. The whole protocol architecture is first described. The PHY layer, MAC layer and NWK layer protocols are then explained in sequence. After that, the data correlation in WSNs is investigated based on both the numerical analysis and literature survey. The data correlation analysis is conducted based on the practical dataset and analysis results show that both spatial and temporal correlations exist in the collected sensor data. Sections 2.3 and 2.4 introduce two mathematical models that are used in the later discussions in the thesis. Both descriptions of the models and related works are given.

Chapter 3

Energy-Efficient Scheduling Mechanism for Indoor WSNs

3.1 Introduction

Wireless sensor networks (WSNs) have gained increasing popularity in many applications [35]. Currently, wireless sensor networks have been applied in many different scenarios. Take the indoor environments as an example. WSN-based building automation systems can be used to monitor and control heating, air-conditioning and other physical parameters in modern buildings. Due to the inherent technical challenges of WSNs, particularly power consumption and accuracy of data collection, researchers from both academia and industry spare no effort on the research and development of the technology.

Considering the complex signal propagation environments in indoor application, more sensor nodes than necessary are usually deployed in order to ensure the network coverage [36]. Due to the high density of node deployment, sensor data generated from neighbor nodes is highly correlated. The strong correlation leads to data redundancy in WSNs. Meanwhile, the transmission and processing of redundant sensor data consume extra energy [37]. Since the energy of a sensor node is limited by its battery, reducing energy consumption and prolonging the network lifetime become key issues in WSNs. Given the high data correlation, it is possible to improve energy efficiency by reducing the redundant data transmission [38]. Specifically, some nodes can be selected as representatives to generate and transmit data instead of using all the nodes within the network at the same time.

In terms of representative nodes selection in WSNs, there are already several works focused on this area in recent years. Based on the spatial correlation model [38], a theoretical framework is proposed so that the sensing field can be partitioned into smaller areas with high correlation [39]. In each correlation area, a representative node is selected by an iterative node selection (INS) algorithm. Similarly, an α-local spatial clustering algorithm has been proposed in [40]. By definition, the sensor nodes are clustered by two metrics, Euclidean distance of sensor data and predefined communication radius α. In each cluster, a representative node is the one with the highest correlation with all the other nodes within the same cluster. The data reliability of the α-local algorithm is further improved in [41] by proposing a new data correlation model, termed as data density correlation degree (DDCD).

In the aforementioned algorithms, only the representative nodes generate and transmit data all the time. Hence, there is a potential problem that the representative nodes run out of battery faster than other nodes. The dead representative nodes can affect the connectivity of the network. Therefore, it is necessary to balance the energy consumption of nodes within the whole network. In the literature, two types of energy-balanced methods are proposed, *i.e.*, energy-aware spatial correlation mechanism [42] and sensor scheduling mechanism [43].

In order to balance the energy consumption, the residual energy of a node is considered as a new metric when selecting the representative node [42]. Additionally, both the correlation areas and the representative nodes are adaptively updated so that the network performance can be further improved. By contrast, a different energy balance approach is proposed in [43]. In the latter work, the sensor nodes are clustered based on the spatial correlation of sensor data first. Different from the representative selection algorithms, no special representative node is selected while every node within the cluster takes turns to be the representative. The biggest problem in these two works ([42] and [43]) is how to make sure of network connectivity. That is why the ZigBee cluster tree topology is adopted in this work.

In this chapter, a new spatial correlation based sensor scheduling mechanism is proposed so that the network energy efficiency of indoor WSNs can be improved. First, a new cluster formation algorithm is proposed, termed as adaptive dual-metric K-means (adaptive DK-means) algorithm. The new clustering algorithm exploits both data correlation between multivariate sensor data and ZigBee address as clustering metrics, which improves the data reliability of generated groups. Then a new sensor scheduling algorithm is designed to determine the schedule of nodes acting as group representatives. At last, a brief updating algorithm is proposed to adaptively adjust the groups.

The simulations are conducted in OPNET with the dataset published by Intel Berkeley research lab [7]. Simulation results demonstrate that the new cluster formation algorithm decreases the average relative error by 73.6% with a 0.8% increment in simulation time, as compared to the adaptive K-means algorithm. Additionally, simulation results also show that the new sensor scheduling algorithm reduces the energy consumption by 57.9%, as compared to the baseline ZigBee routing protocol.

The remainder of this chapter is organized as follows. Section 3.2 introduces some basic models used in this work, namely, propagation model and energy consumption model. All the proposed algorithms are explained in detail in Section 3.3. Section 3.4 presents the performance evaluation of the proposed algorithms conducted in OPNET. Finally, conclusions are drawn in Section 3.5.

3.2 System Model

3.2.1 Propagation Model

Considering the multiple obstacles, indoor environment is much more complex than the open areas for wireless communication. Based on the study of indoor signal propagation, a multi-wall model (MWM) has been proposed in the final report of the European Co-Operation in the field of Scientific and Technical research Action 231 (COST-231) [44]. Here, the MWM is adopted as the indoor propagation model (Fig.3.1).

The signal attenuation in MWM is calculated as [44]

$$PL_i(d) = 20log_{10}(\frac{4\pi d}{\lambda}) + k_f^{(\frac{k_f+2}{k_f+1}-0.46)} L_f + \sum_{i=1}^{N} k_{wi} L_{wi}, \tag{3.1}$$

where d is the distance between the receiver and the transmitter. λ is the wave length. L_f refers

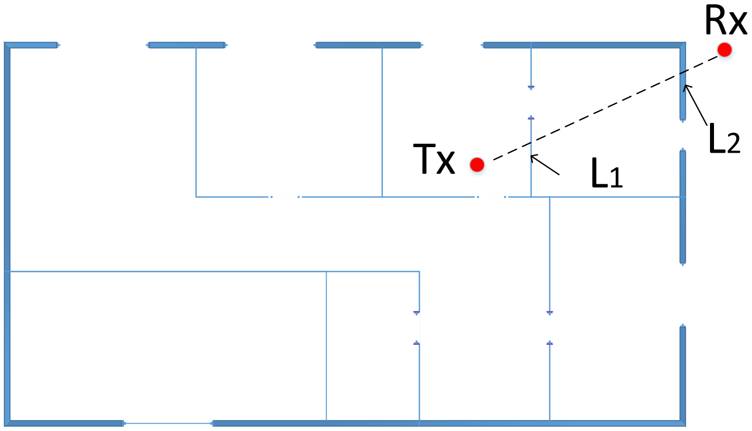


Figure 3.1: Multi-wall model in COST-231.

to the signal strength attenuated by the floor, and k_f is the number of floors traversed by the signal. The signal strength attenuated by the wall with type i is termed as L_{wi}, while k_{wi} is the number of traversed walls with type i. The attenuation values of concrete floor and different walls are shown in Table 3.1 [45].

Table 3.1: Attenuation Values of Different Materials

Material	dB	Material	dB
Type i: Wooden	12.3	Type ii: Stucco	13.1
Type iii: Concrete	16	Concrete Floor	29

3.2.2 Energy Consumption Model

The battery model of MICAz [46] is adopted as the energy consumption model. The currents at different states are shown in Fig.3.2.

The total energy consumption (E) at different states is calculated as

$$E = V(I_{Tx}t_{Tx} + I_{Rx}t_{Rx} + I_{Idle}t_{Idle} + I_{Sleep}t_{Sleep}), \tag{3.2}$$

where V is the voltage, $I_{Tx}, I_{Rx}, I_{Idle}, I_{Sleep}$ are the currents of the transmitting, receiving, idle and sleeping states, respectively. $t_{Tx}, t_{Rx}, t_{Idle}, t_{Sleep}$ denote the time spent in each of the above mentioned states.

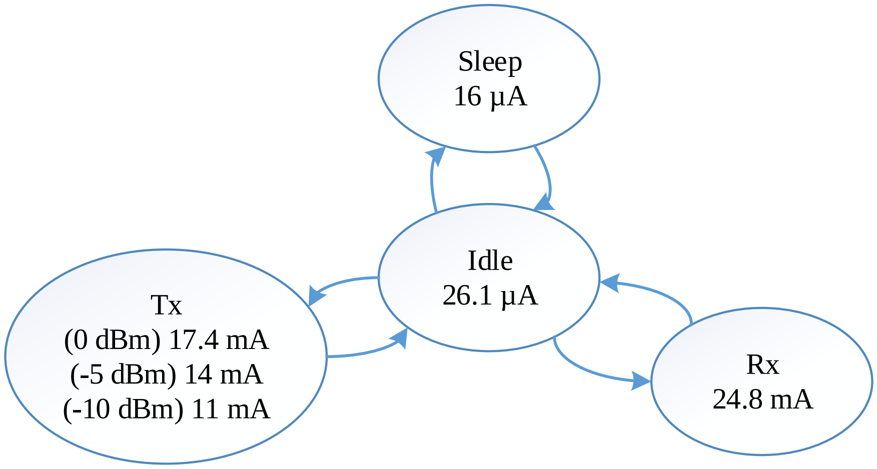


Figure 3.2: Currents at transmitting (Tx), receiving (Rx), idle and sleeping states.

3.3 Cluster Formation and Sensor Scheduling Algorithms

This section presents our proposed cluster formation and sensor scheduling algorithms. First of all, the network is set up under the rule of IEEE 802.15.4/ZigBee protocol stack (more details in Chapter 2.1). Then the adaptive DK-means algorithm is implemented by the PAN coordinator to cluster the sensor nodes into groups. After that, the sensor nodes falling in the same group are scheduled to be group representative and send data back to the base station. The working schedules are decided by the sensor scheduling algorithm. Only if the relative data difference between two consecutive representatives within the same group is larger than a threshold, the PAN coordinator updates the groups by re-operating the adaptive DK-means algorithm. The procedures are organized into a flowchart, as shown in Fig.3.3.

3.3.1 Adaptive DK-means Algorithm

In this section, a new adaptive dual-metric K-means (DK-means) algorithm is presented, where both ZigBee address and the multivariate sensor data are adopted as clustering metrics. More specifically, the algorithm works as follows:

1. Randomly select one node from the sensor nodes set S to be the centroid of group j.

2. If node i satisfies conditions i and ii, it will be removed from S and assigned to group j.

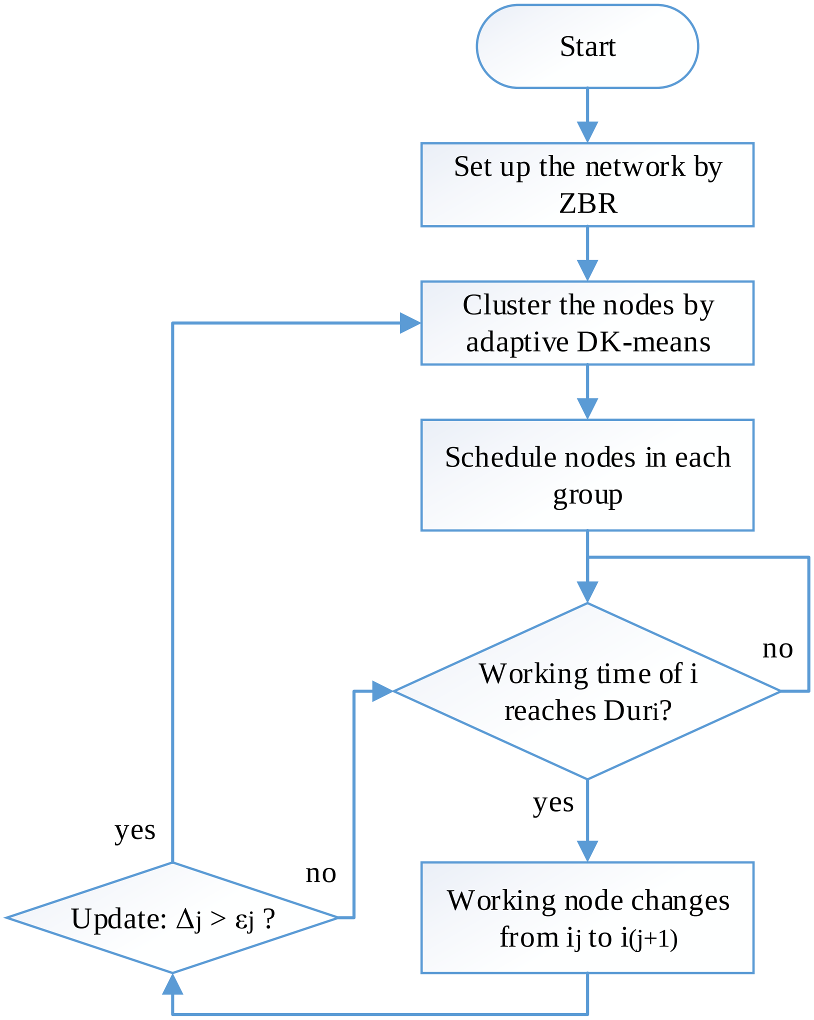

Start
Set up the network by ZBR
Cluster the nodes by adaptive DK-means
Schedule nodes in each group
Working time of i reaches Duri?
no
yes
Working node changes from ij to i(j+1)
Update: Δj > εj ?
yes
no

Figure 3.3: Flowchart of the proposed schemes in the sensor scheduling mechanism.

i. The distance from the ZigBee address of node i to that of group j centroid is below the predefined threshold α. It is mathematically presented as

$$|A_i - CA_j| \leq \alpha, \tag{3.3}$$

where A_i is the ZigBee address of node i and CA_j is average ZigBee address of group j.

ii. The average Euclidean distance of multivariate sensor data between node i and the centroid of group j is below the predefined threshold β. Mathematically,

$$\frac{1}{P}\sum_{p=1}^{P} \|X_{ip} - C_{jp}\| \leq \beta, \tag{3.4}$$

where X_{ip} refers to the p^{th} row in the data matrix of node i (X_i) and C_{jp} indicates that of group j's data centroid (C_j). Specially, the p^{th} row in data matrix can be regarded as an N-dimensional vector. Furthermore, P is the number of physical phenomena that one sensor node monitors. The thresholds α and β in (3.3) and (3.4) are empirical values that are discussed in Section 3.4.

3. Every time a new node joins in group j, the values of group centroid, CA_j and C_j, are updated with the new averages of all the sensor nodes within group j.

4. When there is no more sensor node satisfies conditions i and ii, a new group is set up and the previous procedures are repeated with the remaining nodes in S.

5. Only if the node set S is empty, the cluster formation algorithm will be terminated.

Pseudocode of the proposed algorithm is described in detail in Algorithm 1. Specifically, M refers to the number of sensor nodes, N indicates the data dimension and P stands for the number of physical parameters. Additionally, X_i is node i's data matrix ($P \times N$). NC_j refers to the total number of nodes assigned to group j. IC_i denotes the specific group that node i is assigned to. In other words, $IC_i = j$ means that sensor node i is assigned to group j ($i \in G_j$). Finally, the output K reveals the number of groups that the sensor nodes are gathered into.

Algorithm 1 Adaptive DK-means Algorithm

```
Input:$M$, $N$, $P$, $ITER$, $A$, $X_1, X_2, ..., X_M$
//initialize K and the sensor nodes set S
  $K = 0$;
  $S = \{node_1, node_2, \ldots, node_M\}$;
for t=1,2,...,ITER do
    $K = K + 1$;
  //randomly choose a node n from S as centroid of group K
    $C_K = X_n$, $CA_K = A_n$;   $\forall\ node_n \in S$
  for i=1,2,...,M do
    if $node_i \in S$ then
      if $|A_i - CA_K| \le \alpha$ && $\frac{1}{P}\sum_{p=1}^{P} \|X_{ip} - C_{Kp}\| \le \beta$ then
        //assign node i to group K
          $IC_i = K$, $NC_K = NC_K + 1$;
          update $C_K$, $CA_K$;
          remove $node_i$ from $S$;
      end if
    end if
  end for
  if $S = \emptyset$ then
      output: $K$, $NC$, $IC$;
      break;
  end if
end for
Output:$K$, $NC$, $IC$
```

3.3.2 Sensor Scheduling Algorithm

This subsection describes how the sensor scheduling algorithm works. First, a new concept "super-cycle (SC)" is defined, which is calculated as

$$SC = BI \times lcm\{NC_1, NC_2, \ldots, NC_K\}, \tag{3.5}$$

where *BI* is beacon interval defined in IEEE 802.15.4 MAC layer protocol (mentioned in Section 2.1.1) and *lcm* is short for least common multiple calculation. All the groups within the network follow the same SC.

In each group, the sensor nodes take turns to work as the group representative. Within one SC, the working duration (Dur_i) of node i in group j is given by

$$Dur_i = SC/NC_j, \tag{3.6}$$

and then it sleeps for $(NC_j - 1)SC/NC_j$.

Additionally, the working order (Ord_i) of nodes in each group is decided by the order of their ZigBee addresses. Fig.3.4 shows an example of the sensor scheduling algorithm.

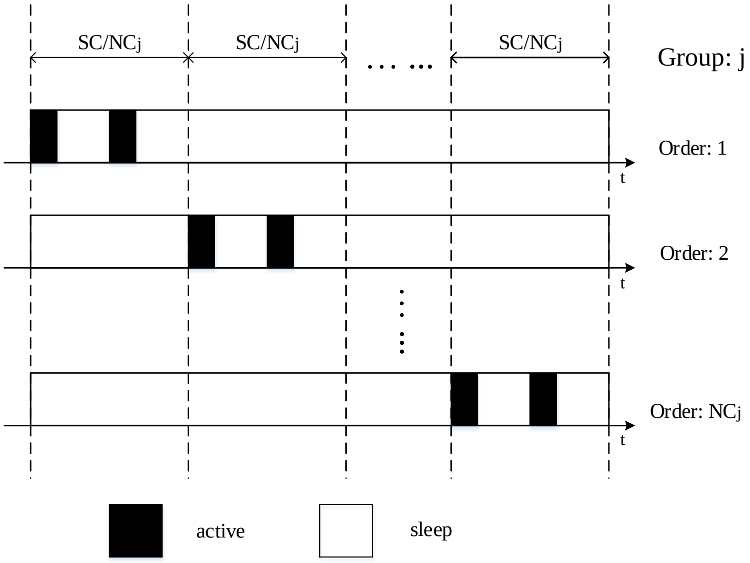


Figure 3.4: Example: sensor scheduling in group j.

Pseudocode of the sensor scheduling algorithm is listed in Algorithm 2. PAN coordinator implements the algorithm and sends the scheduling information (Table 3.2) back to the sensor nodes, and then each sensor node in the network works according to its scheduling information.

Algorithm 2 Sensor Scheduling Algorithm

```
**Input**:$NC$, $IC$, $A$, $K$, $M$
//calculate the super-cycle ($SC$)
$SC = BI \times lcm\{NC_1, NC_2, \ldots, NC_K\}$;
//calculate the duration $Dur_i$
**for** i=1,2,...,M **do**
    $j = IC_i$;
    $Dur_i = \frac{SC}{NC_j}$;
**end for**
//calculate the order $Ord_i$
**for** j=1,2,...,K **do**
    $Ord_i$ = **sort** $i \in G_j$ **by** $A_i$;
**end for**
**Output**:$SC$, $Dur$, $IC$, $Ord$
```

Table 3.2: Scheduling Information

Node ID	Group ID	Order	Duration	super-cycle
i	IC_i	Ord_i	Dur_i	SC

3.3.3 Group Update

In order to make sure the data reliability, the groups of sensor nodes are adaptively updated. More specifically, the PAN coordinator re-runs the cluster formation algorithm to update the groups when the relative data difference (Δ_j) between two consecutive representatives of group j is beyond the threshold (ε_j). The relative data difference Δ_j is given by

$$\Delta_j = \frac{1}{P} \sum_{p=1}^{P} \frac{|X_{i_2 p1} - X_{i_1 pl}|}{X_{i_1 pl}}, \tag{3.7}$$

where $X_{i_1 pl}$ is the last value of parameter p before sensor node i_1 sleeps, $X_{i_2 p1}$ is the first value of parameter p after sensor node i_2 wakes up and P is the number of parameters.

The maximum temporal relative difference (ted_j) between the consecutive data of a node in group j is given by

$$ted_j = \max_{\forall i \in G_j, k=1,\ldots,N-1} (\frac{1}{P} \sum_{p=1}^{P} \frac{|X_{ip(k+1)} - X_{ipk}|}{X_{ipk}}), \tag{3.8}$$

where k is the index of elements in the p^{th} row of data matrix X_i. During the clustering process, the maximum spatial relative data difference (sed_j) between any two sensor nodes within group j is calculated as

$$sed_j = \max_{k=1,\ldots,N,\forall i,i' \in G_j} \left(\frac{1}{P} \sum_{p=1}^{P} \frac{|X_{i'pk} - X_{ipk}|}{X_{ipk}}\right). \tag{3.9}$$

And then the update threshold of group j is calculated as

$$\varepsilon_j = ted_j + sed_j + ted_j sed_j. \tag{3.10}$$

3.4 Performance Evaluation

Performance evaluation of the proposed scheduling mechanism is conducted in OPNET. The specific simulation settings are summarized in Table 3.3 and the network deployment in OPNET is shown in Fig.3.5. In terms of the sensor data, the temperature and humidity sensor readings published by the Intel Berkeley Research Lab [7] are used here.

Table 3.3: Simulation Settings

Number of Nodes (Fig. 3.5)	ZigBee End Devices: 54, ZigBee Routers:17, PAN Coordinator: 1
Area	50m× 50m
PHY Parameters	f=2.4GHz, r=250Kbps, λ=0.125m
MAC Parameters	BO=7, SO=4, $Duty\text{-}cycle$=12.5%
NWK Parameters	C_m=8, R_m=3, L_m=4
Traffic Parameters	Packet Generation: 1 pkt/s Packet Size: 1000 bits/pkt
MICAz Parameters	I_{Tx}=11mA(-10dBm), I_{Tx}=14mA(-5dBm), I_{Tx}=17.4mA(0dBm), I_{Sleep}=16μA, I_{Rx}=27.7mA, I_{Idle}=35μA, V=3V
Simulation Time	300 s

As shown in Table 3.3, the ZigBee network works at the 2.4 GHz frequency band that is the commonly used working frequency of WiFi networks. There is a potential interference between these two techniques, but previous research has proved that WiFi and ZigBee networks can coexit [47].

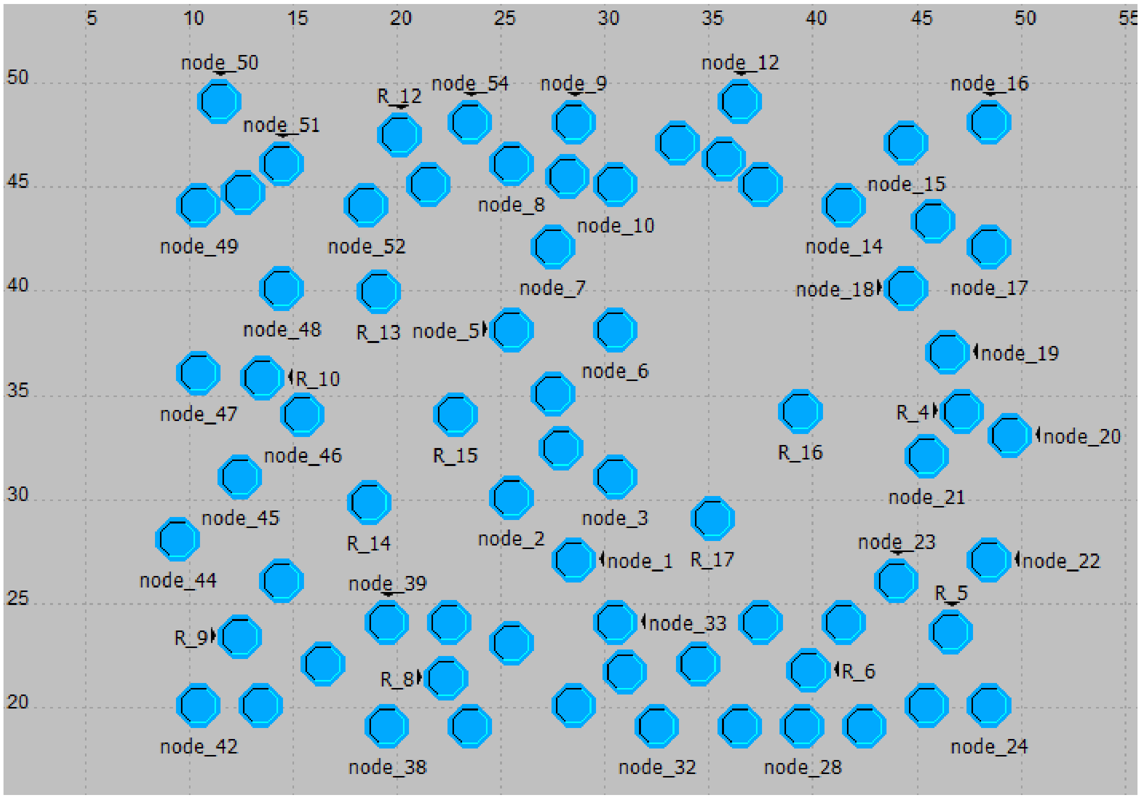


Figure 3.5: Nodes deployment in OPNET.

In Section 3.4, Subsection 3.4.1 analyzes the influence of the thresholds α and β on the total number of groups (K). In Subsection 3.4.2 and 3.4.3, the average relative error of the clusters and the energy saved by the proposed mechanism are discussed, separately.

3.4.1 Number of Groups

Fig.3.6 demonstrates the effect of the thresholds α and β on the total number of groups (K). In the simulation, α ranges from 1 to 5 and β ranges from 0 to 7. From Fig.3.6, it can be concluded that with an increment in α or β, the number of nodes satisfying the thresholds of each group increases correspondingly. As a result, the total number of groups reduces, which is shown in Fig.3.6. Notice that the curves overlap when α equals to $3, 4$ and 5, which indicates that the average difference between the ZigBee addresses of sensor nodes from a given group is no more than 3. The reason behind this phenomenon is that the maximum number of sensor nodes a router can support is 5, due to the simulation setting $C_m - R_m = 5$. Furthermore, the number of groups becomes saturated when β is larger than 6. This is because the average Euclidean distance between sensor data is less than 6. Therefore, when β is larger than 6, the clustering result is mainly affected by α. Thus, the range of α is set to $1 \sim 3$ and β is set to $1 \sim 6$ in the following discussions.

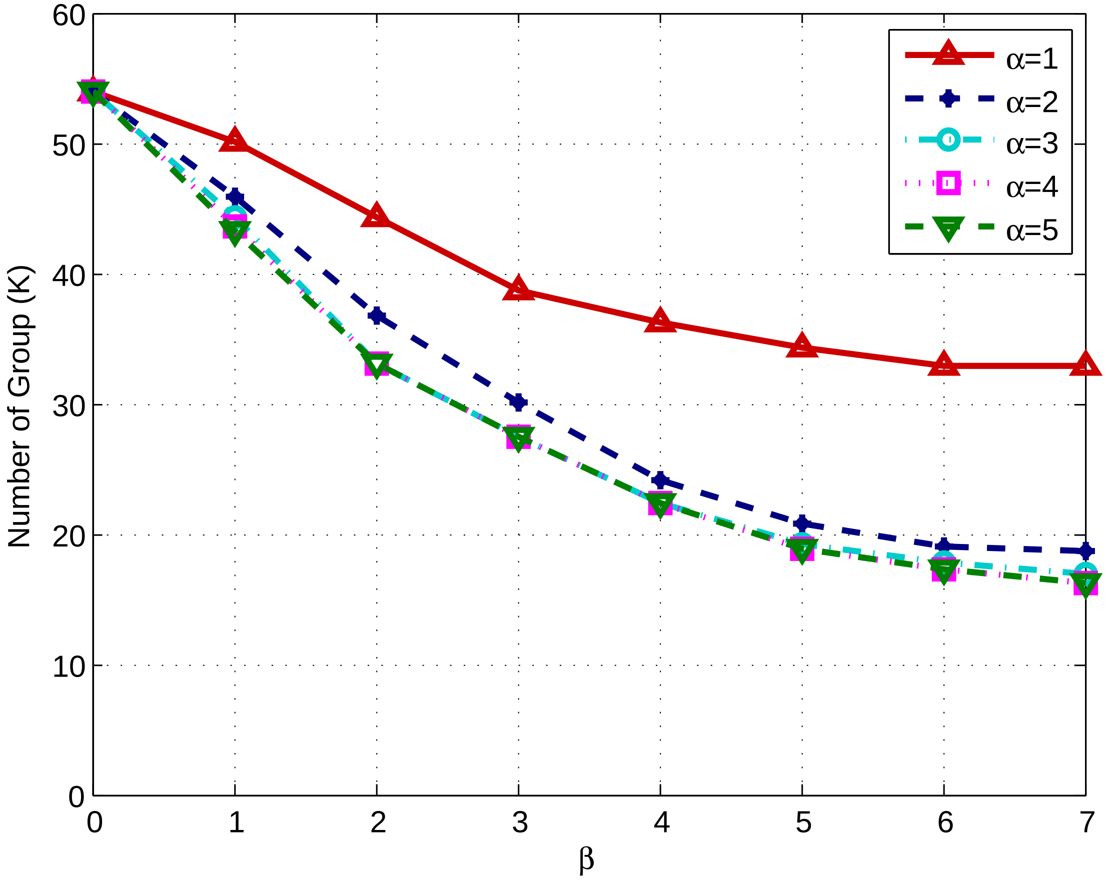


Figure 3.6: Effect of adaptive DK-means algorithm with different α and β on the total number of groups (K).

3.4.2 Average Relative Error

The average relative error (ARE), e_a, is defined as a metric to evaluate the data reliability of the groups that consist of the correlated sensor nodes. More specifically, ARE is the average relative Euclidean distance of data between a given sensor node and its group centroid [41]. ARE for data matrices is derived based on the previous definition.

The relative error of a given node i from group j is calculated as

$$e_i = \frac{1}{P}\sum_{p=1}^{P}\frac{||X_{ip} - C_{jp}||}{||C_{jp}||}, \tag{3.11}$$

where X_{ip} is the p^{th} row in the sensor node i's data matrix, C_{jp} is the p^{th} row in the group j's data centroid matrix. P is the number of the physical parameters. The average relative error of group j is further given by

$$e_j = \frac{1}{NC_j}\sum_{i \in G_j} e_i, \tag{3.12}$$

where $i \in G_j$ indicates that sensor node i is assigned to group j. Besides, NC_j is the total number of nodes assigned to group j. Finally, the average relative error of groups within the whole network is calculated as

$$e_a = \frac{1}{K}\sum_{j=1}^{K} e_j, \tag{3.13}$$

where K is the total number of groups.

The clustering result of the proposed algorithm is compared to that of adaptive K-means algorithm based on ARE, where adaptive K-means algorithm is a classical clustering algorithm considering Euclidean distance of sensor data as the single metric [48].

From Fig.3.7, it can be seen that the ARE of the proposed clustering algorithm is smaller than the adaptive K-means algorithm, which indicates that our proposed algorithm improves the data reliability of the clustering results. This is due to that the adaptive K-means algorithm does not cluster the sensor nodes with the constraint of ZigBee address. The groups consisted of the long-distance sensor nodes lead to the low data reliability. Additionally, given a fixed α, the ARE of the proposed clustering algorithm increases with the growth in β. Similarly, ARE of our proposed algorithm grows with the increment in α when β is fixed. This is because with the increment in α or β, the average number of sensor nodes assigned to a certain group grows. As a consequence, the average Euclidean distance of data grows as well.

The proposed clustering algorithm ($\alpha = 3$) and the adaptive K-means algorithm are simulated 1000 times, separately. The average simulation time is calculated. The calculation results of two algorithms are listed in Table 3.4. Table 3.4 shows that the adaptive DK-means

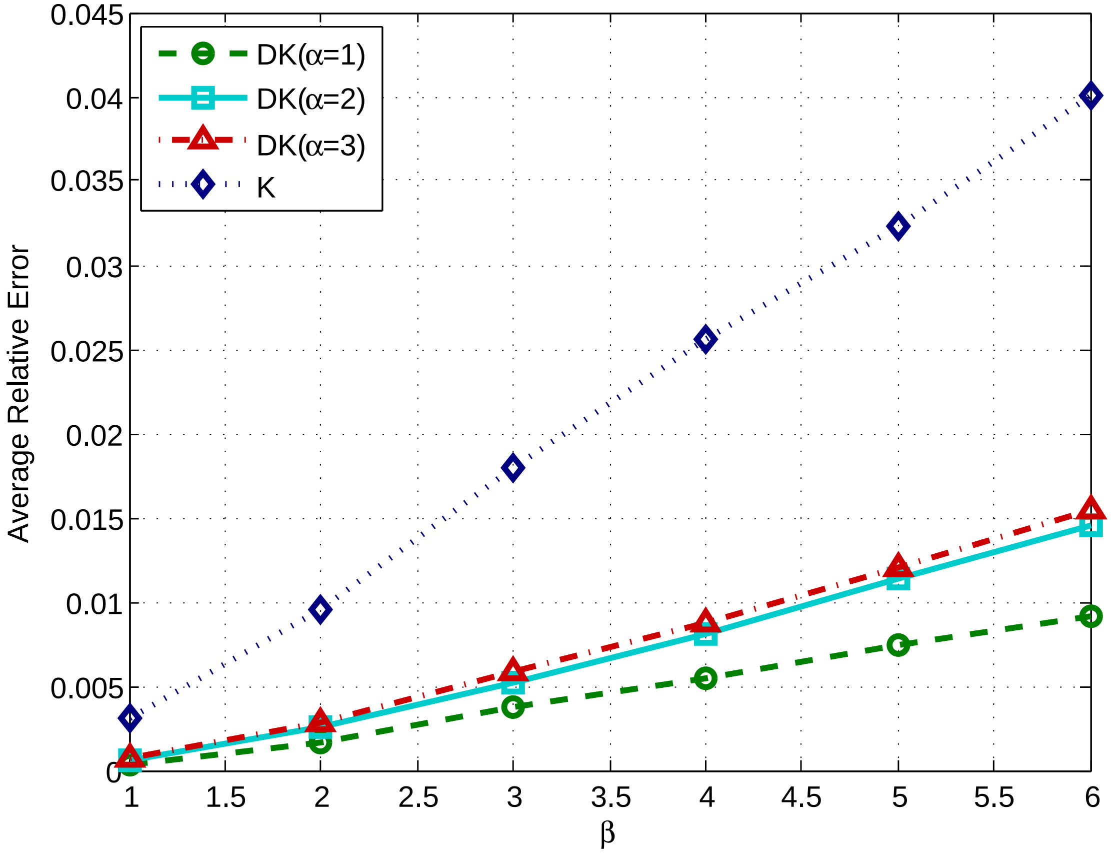


Figure 3.7: Comparisons between adaptive DK-means algorithm with different α and adaptive K-means algorithm with β ranging from 1 to 6.

Table 3.4: Simulation Time (ms)

β	1	2	3	4	5	6
K	49.017	48.814	48.767	48.807	48.735	48.719
DK	49.209	49.34	49.153	49.111	49.182	49.154

algorithm costs about 0.37 ms more simulation time than the classical algorithm on average, which indicates that the proposed clustering algorithm reduces the *ARE* by about 73.6% at the sacrifice of about 0.8% increment in simulation time.

3.4.3 Energy Saving

The energy saved by the proposed scheduling mechanism is evaluated in this subsection. The Open-ZB module [46] is applied in OPNET, so as to conduct the network energy consumption simulations. The energy saving (P_{es}) is given by

$$P_{es} = \frac{E_z - E_s}{E_z} \times 100\%, \tag{3.14}$$

where E_z is the energy consumed by the baseline ZigBee protocol, while E_s is the energy consumed by the proposed mechanism.

Fig.3.8 presents the energy consumption saved by our scheduling mechanism in comparison with the baseline ZigBee protocol. It can be seen that the energy saving increases from 33.7% to 57.9% with an increment in α while β is fixed to 6. Similarly, the energy consumption saved by 18.2% to 57.9% when β increases from 1 to 6 while $\alpha = 3$. These trends in Fig.3.8 are due to that the number of active nodes at the same moment decreases with the increment in α or β, which finally leads to the increased energy savings.

Fig.3.7 and Fig.3.8 jointly show that there is a trade-off between data reliability and energy consumption. If $\alpha = 3$, $\beta = 6$, the energy saving is 57.9% while the average relative error is as high as 0.016. However, the average relative error decreases to 0.00034 while the energy saving is as low as 9% when $\alpha = 1$, $\beta = 1$. The trade-off is because redundant data transmissions ensure the data reliability at the sacrifice of extra energy consumption, and vice versa. In practical applications, the values of thresholds (*i.e.*, α and β) are decided by the specific requirements.

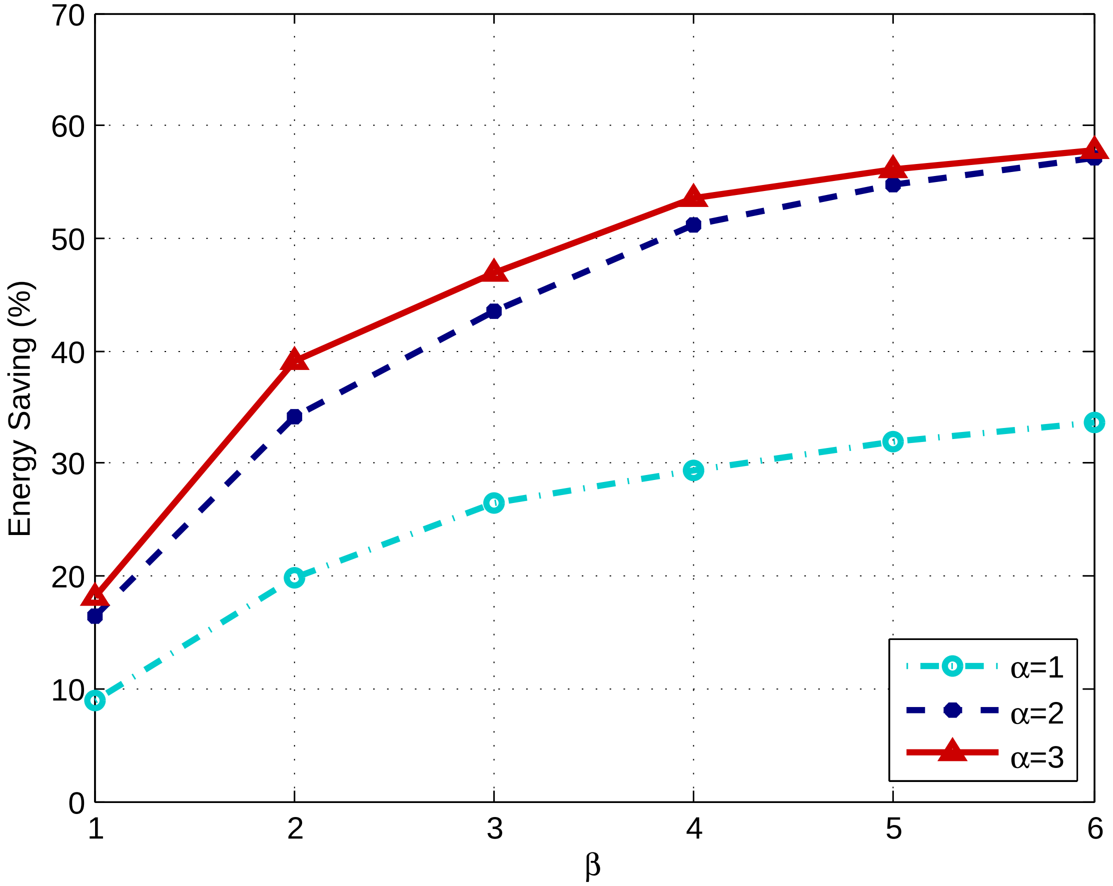


Figure 3.8: Energy saved by the proposed scheduling mechanism compared to the baseline ZigBee protocol with different α and β.

3.5 Chapter Summary

In this chapter, a new sensor scheduling mechanism is proposed in order to improve the network energy efficiency of indoor WSNs. Within the scheduling mechanism, a new cluster formation algorithm is proposed, termed as adaptive DK-means algorithm. The new clustering algorithm is based on the spatial correlation of sensor data. Both ZigBee address and multivariate sensor data are introduced as clustering metrics. Simulation results show that the proposed clustering algorithm improves the data reliability of the clustering results, as compared to the adaptive K-means algorithm. Additionally, a new sensor scheduling algorithm is developed, which is in cooperation with the intrinsic duty cycle in IEEE 802.15.4 MAC protocol. Simulation conducted in OPNET shows that the scheduling mechanism reduces the network energy consumption by up to 57.9%, as compared to the baseline ZigBee protocol.

Chapter 4

Temporal and Spatial Correlation based Distributed Fault Detection Algorithm in WSNs

4.1 Introduction

Due to their self-organizing nature, wireless sensor networks (WSNs) are normally used to achieve long-term monitoring in tough and inaccessible environments. However, long-term operations in harsh environments make the sensor nodes more vulnerable to different kinds of attacks. These attacks could eventually lead to faults in sensor nodes. Generally, faults can be categorized into function fault and data fault [49]. Function fault refers to the malfunctions occurred at certain hardware components, namely, power supply, transceiver and processor. It is easier to detect function faults, since the compromised nodes behave abnormally, like routing failure, packet loss and etc. By contrast, the compromised sensor nodes are hard to notice when the data fault occurs. This is because the compromised nodes still have the capacities of generating and transmitting sensor data, but the sensor data collected is faulty. It is critical to detect the faulty data in real time, as faulty sensor data can mislead the data center with erroneous information.

Data fault detection algorithms in WSNs can be summarized into two major categories, namely, centralized method and distributed method. In the centralized algorithms, the base station (BS) plays a key role. All the sensor readings are gathered and processed at the base station, and then feedback is sent back to sensor nodes. Although the computational capacity of BS is more powerful than sensor nodes, the centralized processing still creates heavy burdens on BS. Besides, the transmission of faulty data and incorrect feedback consumes extra network resources. Therefore, distributed method is considered to be more effective in achieving data fault detection, since the detection algorithm is implemented at each node locally.

In recent years, there are already some works focused on distributed fault detection in WSNs. These works are briefly reviewed here. A distributed fault detection (DFD) algorithm has been proposed based on spatial correlation [50]. In DFD algorithm, neighborhood majority voting is introduced to detect faulty sensor data, where neighbor refers to the sensor node within one hop. In order to improve the detection accuracy in sparse network, an improved distributed fault detection (iDFD) algorithm is proposed by modifying the critical detection criteria in DFD [51]. Similarly, a normal distribution based error function is introduced as the detection criterion in [52], so that the performance of DFD algorithm can be further improved.

The aforementioned algorithms treat the influences of neighbors in a sensor network equally. However, the influences in reality can not be identical since the distances and confidence levels of neighbors are different. In order to further improve the detection accuracy, the different influences of neighbors are considered in the following works. The distance between neighbor nodes is considered as a weight in evaluating the mutual impact among sensors [53]. On the other hand, confidence levels are considered as weights [54][55]. Given these works, the different strength of correlation and potential statuses of neighbor nodes are taken into consideration during the fault detection procedure. Simulation results show that the added weights definitely improve the detection accuracy.

However, the above-mentioned works exploit the spatial correlation only. Recent works show that detection accuracy in a sparse network can be significantly improved when taking advantage of temporal correlation at the meantime [56][57]. Since the changes of physical parameters in natural environments are continuous, the range of coming measurements can be predicted by historical observations. Both works adopt the auto-regressive moving average (ARMA) model as the time series prediction model [56][57]. However, the temporal and spatial detection procedures are conducted independently in these works. Given this fact, integration of temporal and spatial detection algorithms is needed to be further investigated.

In this chapter, a new distributed fault detection algorithm is proposed, which is based on both the temporal and spatial correlation of sensor data. Within the proposed algorithm, Kalman filter based self-detection is implemented first. This result is then introduced into the weighted-median detection. The proposed weighted-median detection considers both the potential status and received signal strength indicator (RSSI) of a neighbor as weights. The status of sensor data is finally decided by the combination of two detection operations. Compared with other distributed fault detection algorithms, the new data fault detection algorithm improves the detection accuracy, especially when the network is sparse.

The remainder of this chapter is organized as follows. Section 4.2 introduces the data fault model used in this work and compares three time series prediction models. The proposed distributed fault detection algorithm is explained in detail in Section 4.3. Simulations conducted in Section 4.4 compares the proposed algorithm with the previous data fault detection algorithms in the literature. Finally, this chapter is summarized in Section 4.5.

4.2 System Model

4.2.1 Data Fault Model

Hardware components of a typical sensor node are shown in Fig.4.1. According to the classification in [50], the components can be categorized into two groups. The first group consists of power supply unit, processor and transceiver . Malfunctions occurred at this group can be noticed easily, since the node stops functioning, neither processing data nor communicating with others. The second group contains sensors and actuators. Differently, the node may behave “normally” when malfunctions occur at sensors or actuators. It still generates and propagates data as always. However, the data it generates is faulty. Under this condition, a faulty node could be difficult to detect. In this chapter, detection of this kind of fault is studied, which is termed as data fault. According to the invisible characteristics, data fault is categorized into the following types [58].

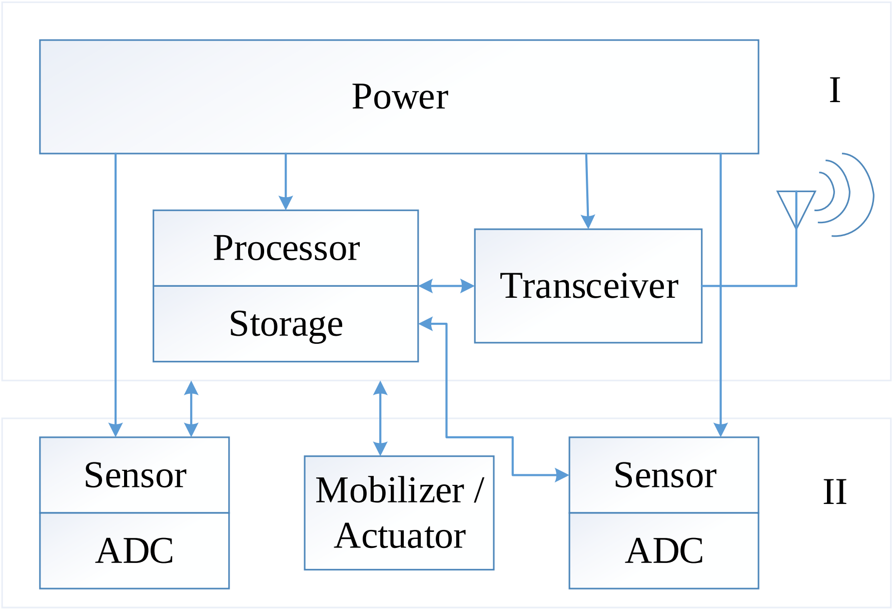


Figure 4.1: Hardware components of sensor node.

- Outlier fault, isolated data fault occurs randomly.
- Spike fault, a series of data outliers occur frequently.
- Stuck-at fault, the sensor readings stay constant without any variance for a certain period.
- Noise fault, a series of faulty data exhibits unexpectedly high variation and may or may not track the normal trend.

Features of the four types of data fault are shown in Fig.4.2, which is plotted with the practical temperature sensor readings sampled at Grand-St-Bernard by the SensorScope group[59].

4.2.2 Network Model

We assume that all the senor nodes are randomly deployed in the monitored areas and homogeneous with the same transmission range. The two-ray path loss model is adopted as the propagation model without losing generality [60].

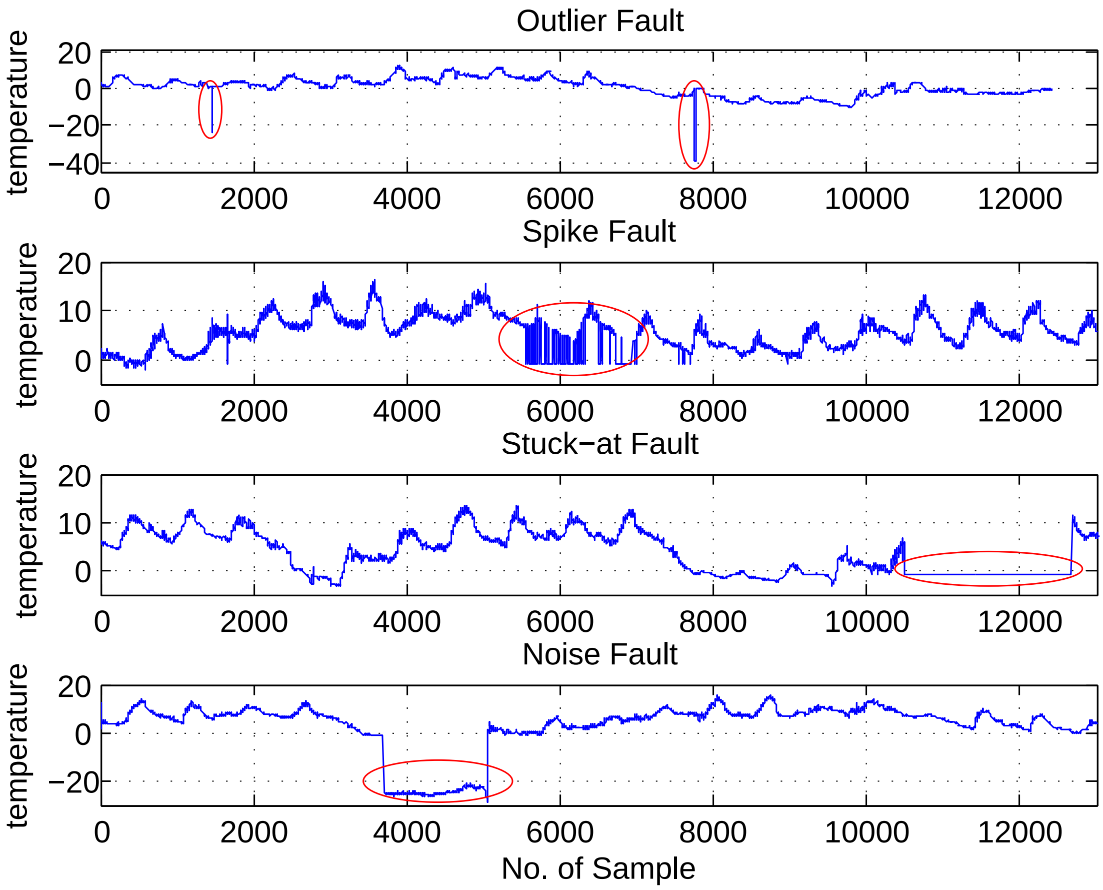


Figure 4.2: Four types of data fault: Outlier, Spike, Stuck-at, Noise.

4.2.3 Time Series Prediction Model

In the proposed algorithm, time series prediction model is used. Since the prediction accuracy is mainly decided by the chosen model, comparisons among different models are necessary. Three time series prediction models, Kalman filter, grey model (GM(1,1)), auto-regressive moving average model (ARMA(1,1)) are considered. The brief descriptions of these models are presented in this section.

4.2.3.1 Kalman Filter

The Kalman filter consists of two major procedures: prediction and update [61]. In the prediction procedure, data at time t ($\hat{X}_{t|t-1}$) is estimated by

$$\hat{X}_{t|t-1} = A_t\hat{X}_{t-1|t-1} + B_tU_t, \tag{4.1}$$

and its corresponding covariance ($P_{t|t-1}$) is calculated as

$$P_{t|t-1} = A_tP_{t-1|t-1}A_t^T + Q_t, \tag{4.2}$$

where A_t is the state transition model, B_t is the control-input model applied to the control vector U_t, Q_t is the covariance of Gaussian white noise in the prediction period. Using (4.1) (4.2), the optimal Kalman gain is calculated as

$$K_t = P_{t|t-1}H_t^T(H_tP_{t|t-1}H_t^T + R_t)^{-1}, \tag{4.3}$$

where H_t is observation model and R_t is the covariance of observation noise. Then $\hat{X}_{t|t}$ and $P_{t|t}$ can be updated with the optimal Kalman gain (K_t) and the measured data (Y_t) as

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t(Y_t - H_t\hat{X}_{t|t-1}), \tag{4.4}$$

$$P_{t|t} = (I - K_tH_t)P_{t|t-1}. \tag{4.5}$$

Similarly, $\hat{X}_{t|t}$ and $P_{t|t}$ will be used to predict $\hat{X}_{t+1|t}$ and $P_{t+1|t}$. Benefiting from using the Kalman filter, the coming data can be predicted with only the latest sensor reading.

4.2.3.2 GM(1,1)

With the GM(1,1) model, a forthcoming measurement can be predicted with the last t historical observations [61]. The historical data series used to do the prediction is denoted as

$$X^{(0)} = x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(t). \tag{4.6}$$

The 1-AGO (accumulated generating operator) sequence of $X^{(0)}$ ($X^{(1)}$) is denoted as

$$X^{(1)} = x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(t). \tag{4.7}$$

The differential equation of the GM(1,1) model is given by

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b, \tag{4.8}$$

where $[a, b]^T = (B^T B)^{-1} B^T A$. While, $A = \begin{pmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \cdots \\ x^{(0)}(t) \end{pmatrix}$ and $B = \begin{pmatrix} z^{(1)}(2) & 1 \\ z^{(1)}(3) & 1 \\ \cdots & \cdots \\ z^{(1)}(t) & 1 \end{pmatrix}$, where $z^{(1)}(i) = -0.5x^{(1)}(i) - 0.5x^{(1)}(i-1),\ i = 2, 3, \ldots, t$. Therefore, $\hat{x}^{(1)}(t+1)$ is calculated as

$$\hat{x}^{(1)}(t+1) = e^{-ak}\left[x^{(0)}(1) - \frac{b}{a}\right] + \frac{b}{a}. \tag{4.9}$$

Finally, the forthcoming data predicted by $X^{(0)}$ is calculated as

$$\hat{x}^{(0)}(t+1) = e^{-ak}\left[x^{(0)}(1) - \frac{b}{a}\right](1 - e^{a}). \tag{4.10}$$

4.2.3.3 ARMA(1,1)

ARMA(p,q) refers to the combination of auto-regressive model with term p and moving average model with term q [57]. Based on this model, the data value at time t is estimated with

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ldots - \theta_q a_{t-q}, \tag{4.11}$$

where the parameters $\phi_1, \phi_2, \ldots, \phi_p$ and $-\theta_1, -\theta_2, \ldots, -\theta_q$ are estimated by the exact maximum likelihood computational method. Since the first items are the most important, $p = 1$ and $q = 1$ are adopted in the following comparisons.

4.2.3.4 Comparisons

Three models are trained with the practical temperature sensor data provided by Intel Berkeley research lab [7]. The data used here is the first 700 temperature readings (100 samples for training and 600 for testing) from No.1 sensor node sampled every 30s on 2/28/2004. The values of the practical data and the data estimated by the three models are shown in Fig.4.3.

The fitness of a model is evaluated by both the residual error and the mean absolute percentage error. The residual error is the difference between the measured data (x_i) and the estimated data ($\widehat{x_i}$), *i.e.*, $x_i - \hat{x}_i$. The mean absolute percentage error ($MAPE$) is defined as

$$e = \frac{1}{n} \sum_{i=1}^{n} \frac{|x_i - \widehat{x_i}|}{x_i} \times 100\%. \tag{4.12}$$

The residual error of three models are shown in Fig.4.4. It can be seen that the residual error curves of all the three models fluctuate around zero, which implies that all these models match the trend of the data. The fitness of model can be evaluated further with $MAPE$. The $MAPE$ of three models are demonstrated in Table 4.1. It shows that the $MAPE$ of Kalman filter is the lowest, which indicates that it matches the trend of data series the best. Therefore, the Kalman filter is adopted as the time series prediction model in this work.

Table 4.1: Mean Absolute Percentage Error

	Kalman Filter	GM(1,1)	ARMA(1,1)
MAPE (%)	0.121157	0.191773	0.130401

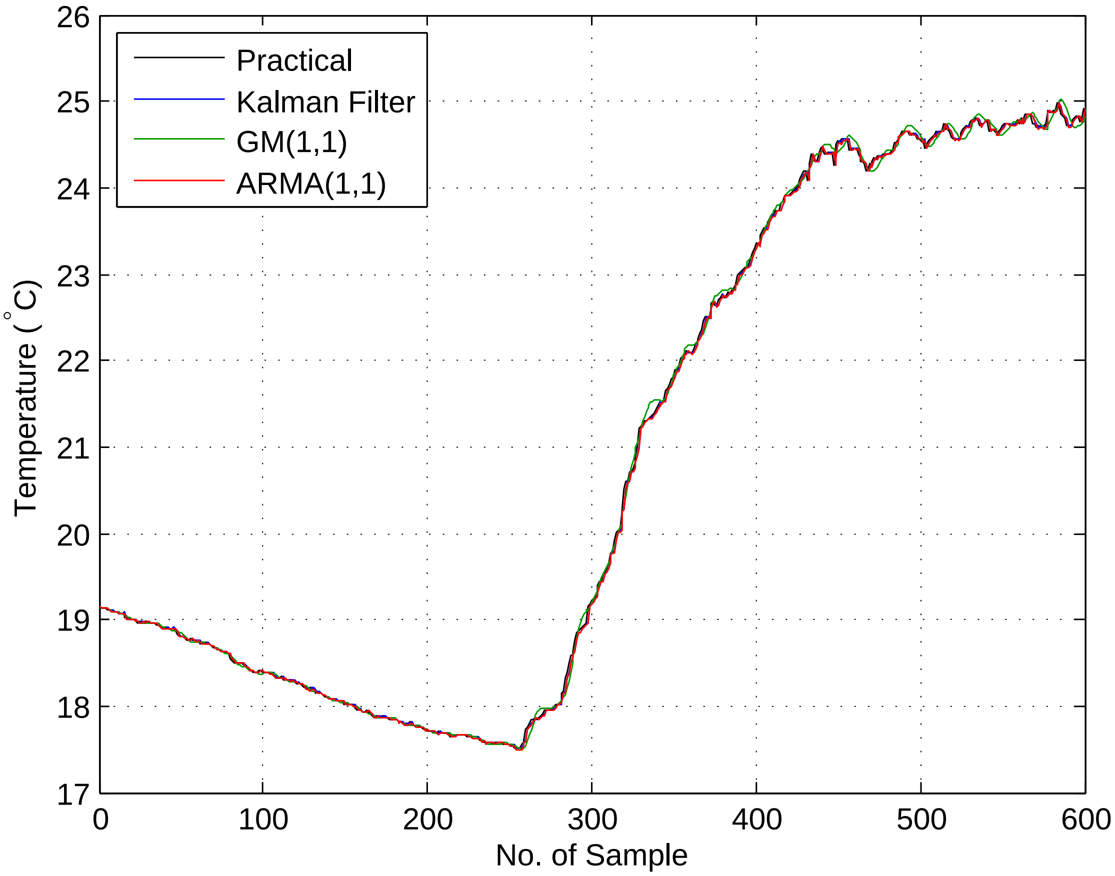


Figure 4.3: Practical data and data estimated by Kalman filter, GM(1,1), ARMA(1,1).

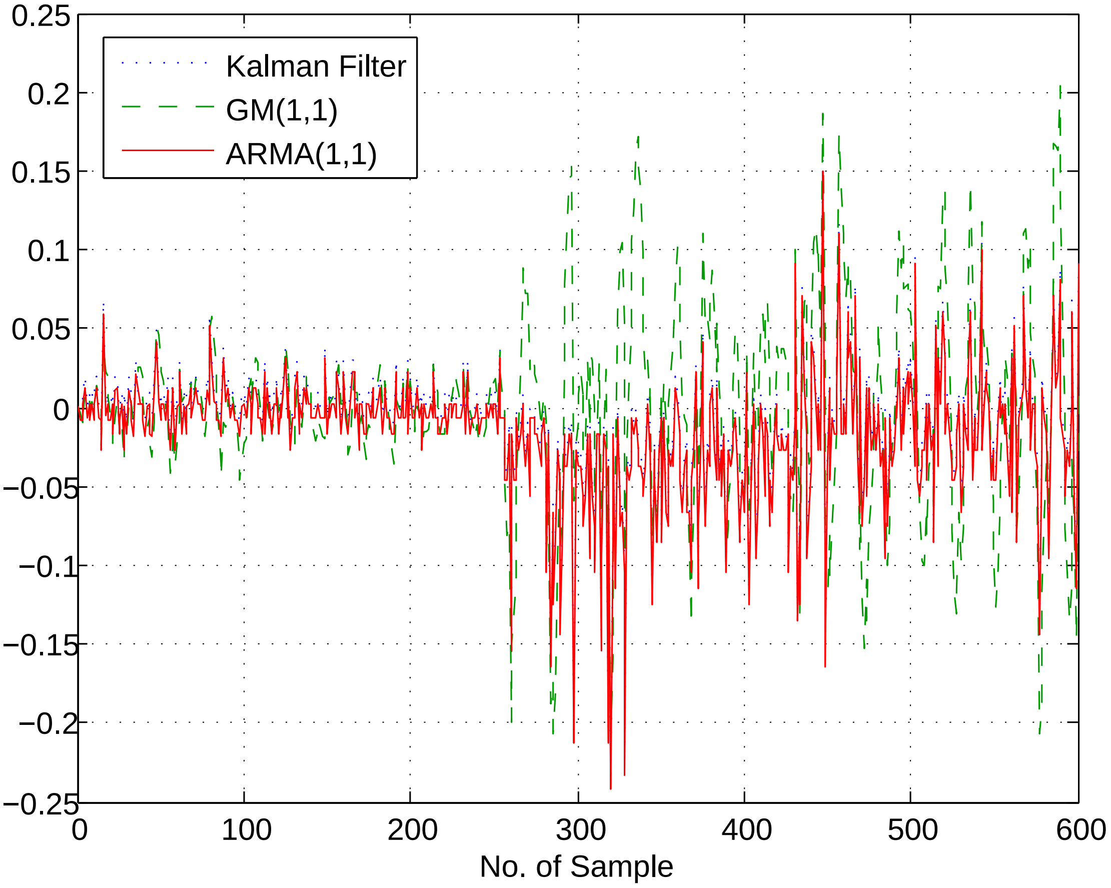


Figure 4.4: Residual error of the 600 samples given by Kalman filter, GM(1,1), ARMA(1,1).

4.3 Distributed Fault Detection Algorithm

The proposed distributed fault detection algorithm, *i.e.*, *TSC*, consists of two main detection procedures: self detection and weighted-median detection. The self detection procedure is based on the temporal correlation of the time series of sensor data. The details of self detection are described in Subsection 4.3.1. Weighted-median detection relies on the spatial correlation between the neighbor sensor data, which is explained in Subsection 4.3.2. If both detection procedures recognize the sensor data as “likely faulty (LF)”, it will be diagnosed as “faulty (FT)” and discarded. A flowchart of the proposed algorithm is demonstrated in Fig.4.5, where N is the total number of sensor nodes in the network.

4.3.1 Self Detection

In the self detection procedure, diagnosis of the measured sensor data at time $t+1$ (x_{t+1}) depends on the comparison with the predicted value ($\hat{x}_{t+1}$). Specifically, $\hat{x}_{t+1}$ is predicted by the Kalman filter as mentioned in Subsection 4.2.3, based on the historical sensor readings already detected as “good (GD)”. If the difference between the predicted value ($\hat{x}_{t+1}$) and the measured sensor reading (x_{t+1}) is beyond a threshold (θ_1), *i.e.*, $\Delta_1 = |x_{t+1} - \hat{x}_{t+1}| > \theta_1$, x_{t+1} will be labelled as “likely faulty (LF)”. Otherwise, it will be labelled as “likely good (LG)”.

4.3.2 Weighted-median Detection

Due to the high density of sensor node deployment, sensor data from spatially nearby sensor nodes is highly correlated. Based on the spatial correlation of sensor data, weighted-median detection method is proposed. The specific procedures of the method are listed below.

1. After self detection, sensor nodes are labelled as either “LG” or “LF”. The to be diagnosed node i collects data $\{x_{i_1}, x_{i_2}, \ldots, x_{i_j}, \ldots\}$ from its “LG” neighbors.

2. During wireless communication procedure, node i can assess and record the RSSIs of the signals transmitted by its “LG” neighbors. Then the weight (λ_{i_j}) of the neighbor data is determined by its RSSI order. For example, if the signal strength of neighbor i_1 is the weakest, then $\lambda_{i_1} = 1$. Similarly, if i_2 is the second weakest, then $\lambda_{i_2} = 2$, and so on.

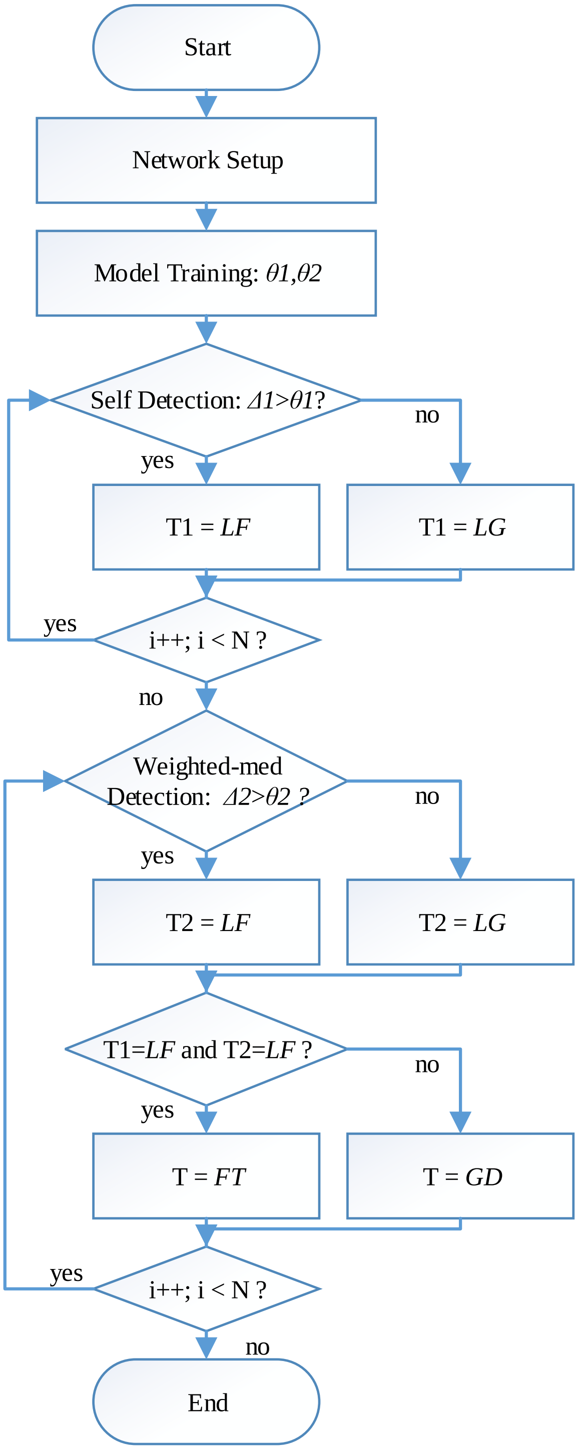


Figure 4.5: Flowchart of the proposed distributed fault detection algorithm, *TSC*.

3. A new data vector is generated by making λ_{i_j} copies of x_{i_j}, *i.e.*, $\{\underbrace{x_{i_1},\ldots,x_{i_1}}_{\lambda_{i_1}},\underbrace{x_{i_2},\ldots,x_{i_2}}_{\lambda_{i_2}},\ldots,\underbrace{x_{i_j},\ldots,x_{i_j}}_{\lambda_{i_j}},\ldots\}$. After that, the new data vector is sorted in ascending order. Median value of the sorted data vector is termed as $\hat{x}_i$.

4. The relative difference between the median value ($\hat{x}_i$) and the sensor data of node i (x_i) is calculated as $\Delta_2 = \frac{|x_i - \hat{x}_i|}{\hat{x}_i}$. x_i is labelled as "LF", only when $\Delta_2 > \theta_2$. Otherwise, x_i is labelled as "LG".

Sensor data x_i is diagnosed as "FT" and discarded, only when it is labelled as "LF" twice, *i.e.*, at both self detection and weighted-median detection procedures. Otherwise, it is diagnosed as "GD" and sent to the base station.

4.3.3 Detection Thresholds

As mentioned in Subsections 4.3.1 and 4.3.2, two detection thresholds are used in the algorithm, *i.e.*, θ_1 and θ_2. Flowchart (Fig.4.5) shows that the detection thresholds are decided during the model training procedure, while how to decide θ_1 and θ_2 is explained in this subsection.

Considering a fit prediction model (discussed in Subsection 4.2.3), the residual error follows the zero-mean Gaussian distribution, *i.e.*, $(x_t - \hat{x}_t) \sim N(0, \sigma^2)$. Hence, the standard deviation (σ) of residual error is calculated as

$$\sigma = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2}, \tag{4.13}$$

where T is the number of samples used to train the time series prediction model. Given a Gaussian distribution (μ is the mean and σ is the standard deviation), the probability of a value within the range $[\mu - 3\sigma, \mu + 3\sigma]$ is 99.7%, according to the 3σ rule [62]. Based on this rule, threshold θ_1 is calculated as

$$\theta_1 = \max_{\forall\ node_i} |\mu_i \pm 3\sigma_i|, \quad where\ \forall \mu_i = 0. \tag{4.14}$$

Threshold θ_2 is defined as

$$\theta_2 = \max_{t=1,\dots,T}\left(\max_{\forall\ node_i}\left(\max_{j=1,\dots,N_i}\frac{|x_{it} - x_{i_jt}|}{x_{i_jt}}\right)\right), \tag{4.15}$$

where x_{it} and x_{i_jt} are data of node i and its j^{th} neighbor at t^{th} sampling instance, separately. N_i is the number of node i's neighbors.

4.4 Performance Evaluation

Performance evaluation of the proposed algorithm is conducted in MATLAB, as compared to two conventional distributed fault detection algorithms. One is the spatial correlation based iDFD [51], the other is the temporal-spatial correlation based TSA∩NV algorithm [57].

4.4.1 Evaluation Metrics

In terms of the performance evaluation of fault detection algorithms, detection accuracy and false alarm ratio are commonly used as the evaluation metrics [50].

4.4.1.1 Detection Accuracy

Detection accuracy is defined as the ratio of the number of faulty readings diagnosed as faulty successfully to the overall faults.

4.4.1.2 False Alarm Ratio

False alarm ratio is referred to the ratio of the number of good readings diagnosed as faulty by mistake to the overall good sensor readings.

4.4.2 Simulation Scenarios

In order to evaluate the performance of the proposed algorithm, simulations based on both practical and synthetic datasets are conducted in MATLAB. The practical dataset is the temperature readings of 54 sensor nodes in Intel Berkeley research lab sampled on 2/28/2004 [7].

The deployment of sensor nodes is shown in Appendix A.1. Additionally, two synthetic scenarios are set up as well, in order to compare the algorithm performances at different network densities. More specifically, Scenario I generates a sparsely distributed network where each node is surrounded by 5 neighbors on average. By contrast, Scenario II generates a situation where sensor nodes are densely distributed. Nodes are randomly distributed in both scenarios. The sensor readings are generated based on the statistical characteristics of the data in practical experiments [7]. The range of the good sensor readings is 19~20 °C. In both practical and synthetic scenarios, faulty data is generated according to the data fault model given in Subsection 4.2.1 and the faulty values range from 23~24 °C. Faulty sensor nodes are randomly chosen following normal distribution. The node fault probability within the network ranges from 0.05~0.25. The specific simulation settings are listed in Table 4.2.

Table 4.2: Simulation Settings

	Practical Scenario	Synthetic Scenario I	Synthetic Scenario II
Network Density	Sparse	Sparse	Dense
Area	50m×50m	30m×30m	30m×30m
Num Of Nodes	54	50	150
Avg Num Of Neighbors	3	5	13
Node Distribution	Map (A.1)	Random	Random
Path Loss Model	Indoor	Two-ray	Two-ray
Data Fault Types	Fault Model	Fault Model	Fault Model
Fault Probability	0.05~0.25	0.05~0.25	0.05~0.25

4.4.3 Simulation Results

Simulation results in both practical and synthetic scenarios are stated in detail.

4.4.3.1 Practical Scenario

The detection accuracy and false alarm ratio of TSC algorithm in practical scenario are shown in Fig.4.6 and Fig.4.7, as compared to iDFD and TSA∩NV algorithms. It can be seen that the proposed TSC algorithm outperforms the other two distributed fault detection algorithms, which improves the detection accuracy and decreases the false alarm ratio. This is because TSC algorithm integrates temporal detection with spatial detection instead of exploiting spatial

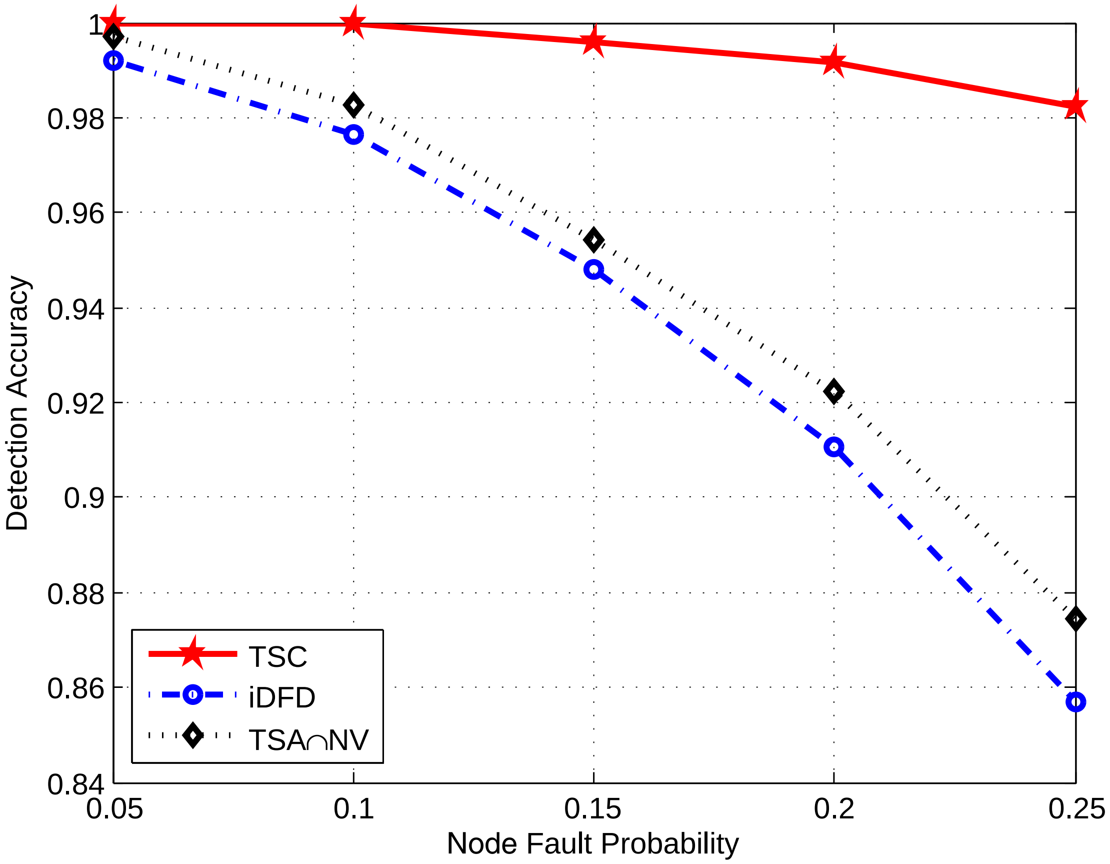


Figure 4.6: Detection accuracy of three algorithms in practical scenario.

detection only (iDFD) or simply uniting the detection results of two procedures (TSA∩NV). Furthermore, the temporal and spatial correlation based TSC and TSA∩NV outperform the spatial only iDFD. The false alarm ratio can be reduced to the minimum by TSC and TSA∩NV, as shown in Fig.4.7. This is because the combined exploitation of temporal and spatial correlation considers two dimensions of data variance so that the detection accuracy can be improved.

Additionally, algorithm performance decreases with the increment in node fault probability. Since all the three algorithms partially or totally rely on the neighbor nodes, high ratio of faulty neighbors can result in false diagnosis of sensor readings.

Simulation time is used to evaluate the time complexity of algorithm. Time consumed by simulations of three algorithms is listed in Table 4.3. The values in Table 4.3 show that the proposed TSC algorithm costs more time than the other two algorithms. It means that TSC improves the detection accuracy while sacrifices the simulation time at the meantime.

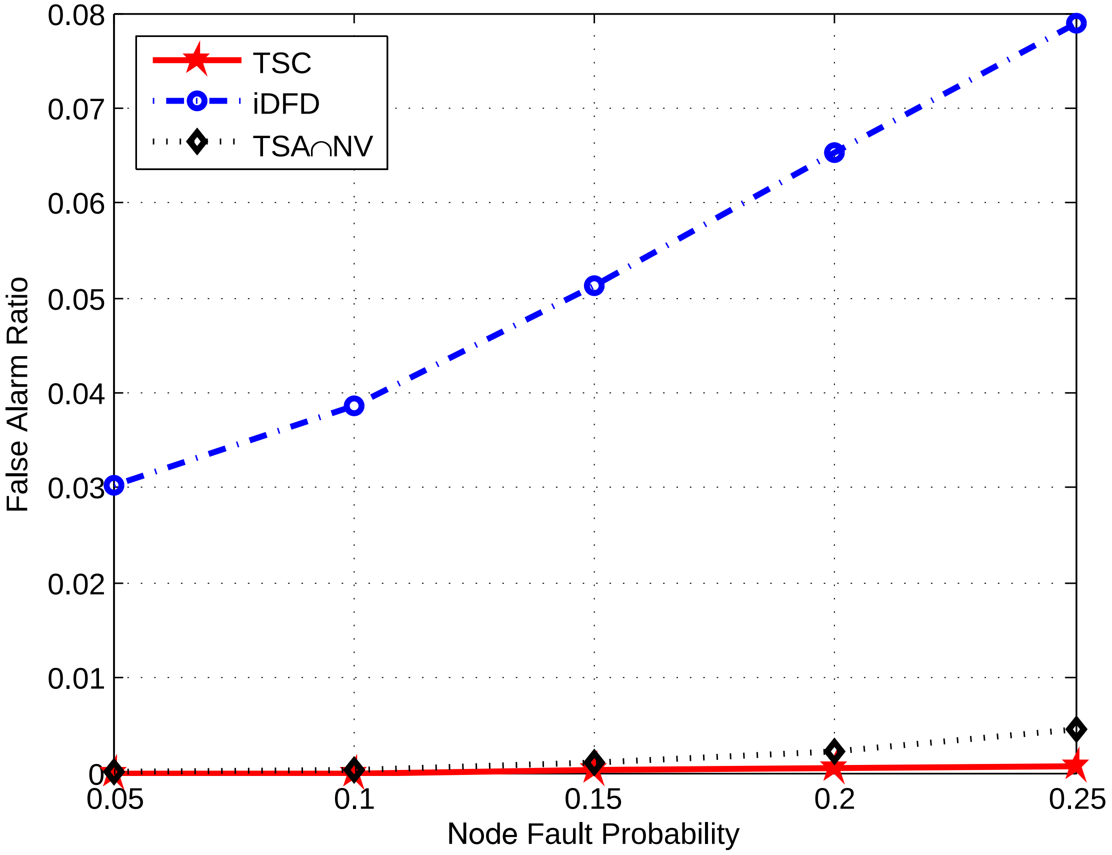


Figure 4.7: False alarm ratio of three algorithms in practical scenario.

Table 4.3: Simulation Time of Practical Scenario (ms)

Algorithm	iDFD	TSA∩NV	TSC
Simulation time	16.0	17.5	20.9

4.4.3.2 Synthetic Scenarios

Fig.4.8 shows an example of deployment of sensor nodes at synthetic Scenario II. It can be seen that nodes are densely distributed in the area and some of them are randomly chosen to be faulty, where black points are the normal nodes while red circles are the faulty nodes.

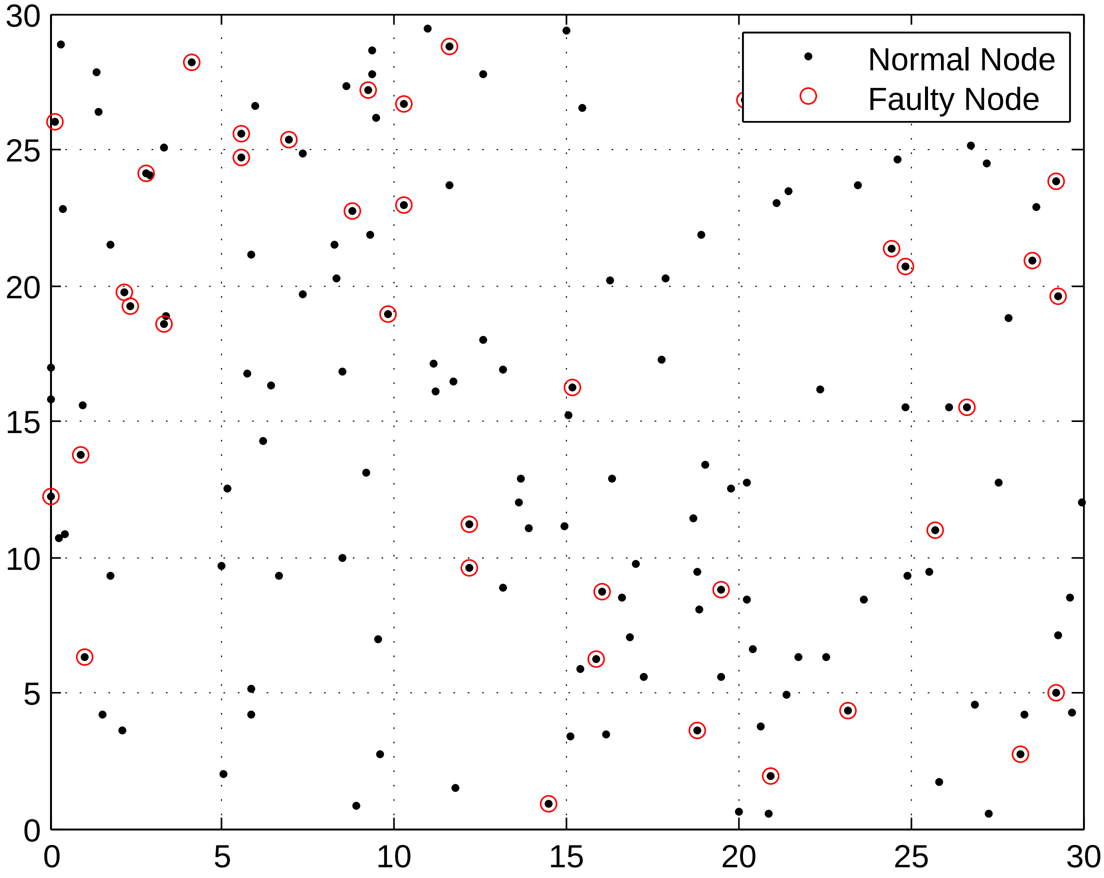


Figure 4.8: Deployment of sensor nodes at synthetic Scenario II, node fault probability = 0.25.

Detection Accuracy Comparisons among TSC, iDFD and TSA∩NV algorithms at two synthetic scenarios on detection accuracy are shown in Fig.4.9.

Similar to the results in Fig.4.6, the detection accuracy of TSC outperforms the other two algorithms. Besides, it is not difficult to notice that the detection accuracy of algorithms at Scenario II is higher than that of Scenario I. The larger average number of neighbors in the dense network finally leads to the preciser diagnosis results, since the algorithms are based on the spatial correlation. Additionally, the detection accuracy of TSC is 100% because the

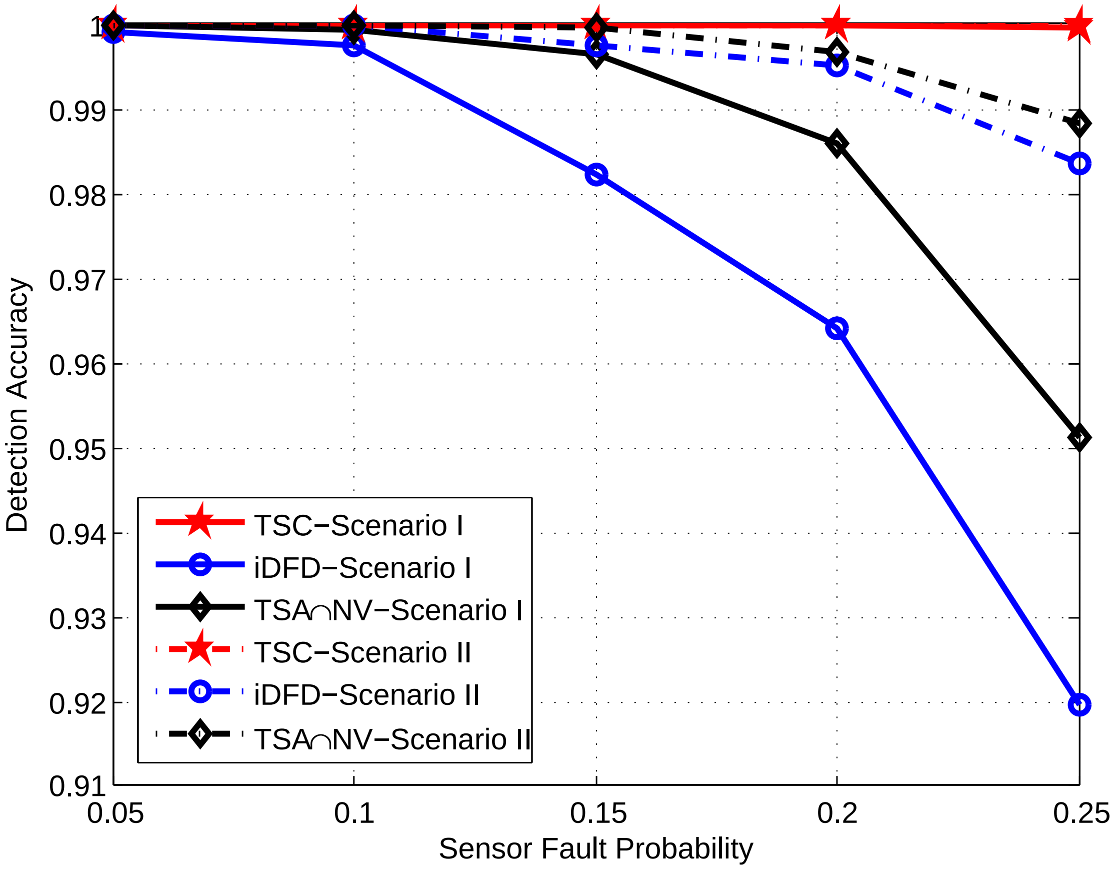


Figure 4.9: Detection accuracy of three algorithms in synthetic scenarios.

synthetic dataset is much more ideal than the practical dataset. The detection accuracy of iDFD and TSA∩NV decreases with the increment in node fault probability. This is due to that these two algorithms do not consider the potential statuses of neighbor nodes during spatial detection so that the detection accuracy outstandingly decreases when the number of faulty neighbors increases.

False Alarm Ratio False alarm ratio of three algorithms is compared in Fig.4.10. It shows that the false alarm ratio of TSC and TSA∩NV is 0%. Meanwhile, the false alarm ratio of iDFD increases from 0.0050 to 0.0269 with the increment in node fault probability at Scenario II. As compared to the range (0.0277~0.0667) at Scenario I, the false alarm ratio decreases. It can be concluded that with the combination of temporal and spatial correlation, the false alarm ratio can be controlled to the minimum. For spatial only detection, decrement in network density not only decreases the detection accuracy, but also increases the false alarm ratio.

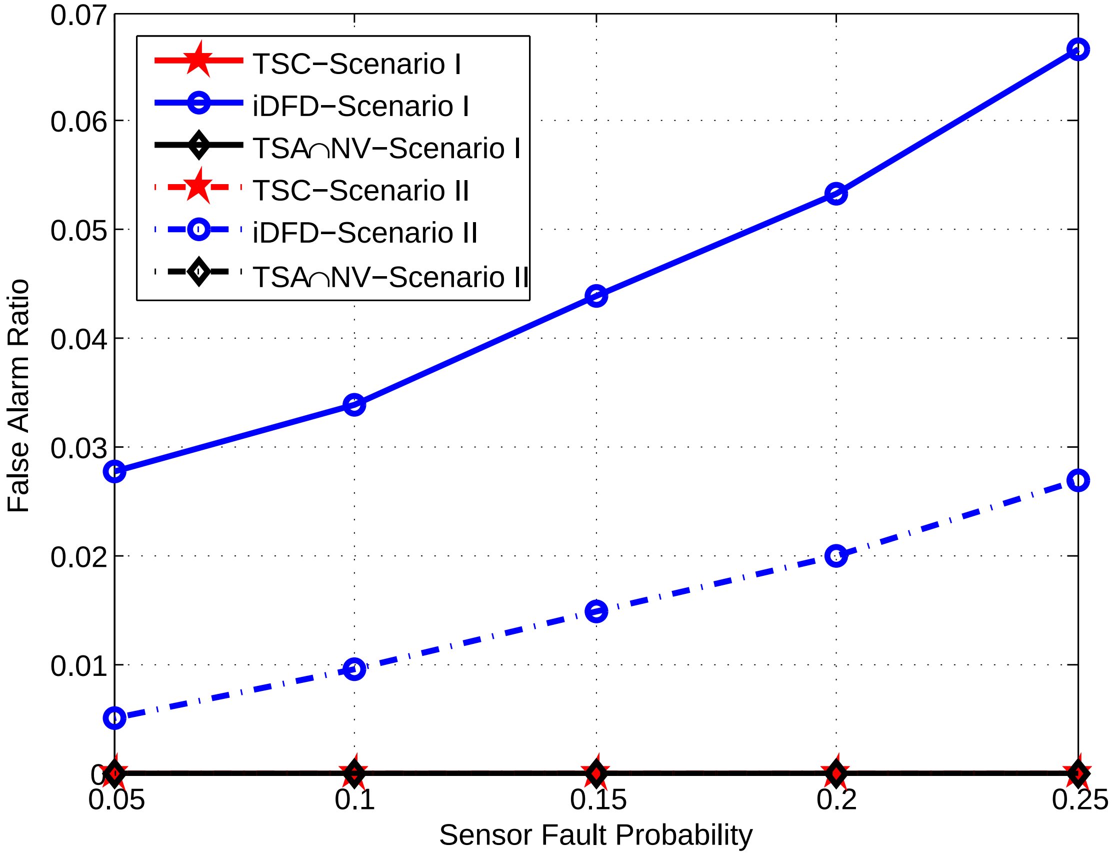


Figure 4.10: False alarm ratio of three algorithms in synthetic scenarios.

Simulation Time According to the data in Table 4.4, TSC algorithm spends more simulation time than iDFD and TSA∩NV at both synthetic scenarios. This is because with the introduction of self detection result and RSSI into weighted-median detection procedure, the time complexity of TSC algorithm is increased.

Table 4.4: Simulation Time of Synthetic Scenarios (ms)

	iDFD	TSA∩NV	TSC
Scenario I	8.1	8.7	10.6
Scenario II	43.7	41.2	52.1

Combined with the fault detection results, it can be seen that the proposed distributed fault detection algorithm outperforms iDFD and TSA∩NV algorithms on detection accuracy and achieves nearly ideal results. However, the achievement of the nearly ideal results costs more simulation time. Therefore, how to balance the algorithm performance and algorithm complexity is worthwhile to be studied in the future.

4.5 Chapter Summary

In this chapter, a new temporal and spatial correlation based distributed fault detection algorithm is proposed. Within the algorithm, the Kalman filter is exploited to predict the collected data from sensor nodes based on the temporal correlation of sensor data first. The actual sensor reading is diagnosed as likely faulty, only if the difference between its value and the predicted one is over a certain threshold. Then the temporal detection result and RSSI based weighted-median detection is further implemented to diagnose the sensor data. The the sensor reading is diagnosed as likely faulty, only when it substantially differs from the weighted median value of its neighbors. The sensor data is detected as faulty and discarded, only if it is detected as likely faulty at both detection procedures. Simulations based on both practical and synthetic datasets are conducted in MATLAB. Simulation results show that the proposed data fault detection algorithm improves the detection accuracy and reduces the false alarm ratio at the cost of more simulation time, as compared to the iDFD algorithm and TSA∩NV algorithm.

Chapter 5

A Novel R-PCA based Data Aggregation Algorithm in WSNs

5.1 Introduction

Due to the rapid progress of wireless sensor networks, a large amount of sensor data has been generated. It is critical to develop new techniques to process the enormous and messy sensor data. Data fault and data redundancy are two major challenges in sensor data processing. As mentioned in Chapter 4, both sensor malfunctions and external interferences can result in faulty sensor data. In terms of data redundancy, it is mainly caused by overfull deployment of sensor nodes. Due to the limited transmission range of sensor nodes and suboptimal node distribution, more sensor nodes than necessary are deployed. The high density of sensor node distribution leads to the highly spatial correlation between sensor data, which finally results in the data redundancy. Given the sensor data correlation, linear correlation based principal component analysis (PCA) can be used to detect the faulty sensor data and aggregate the redundant data. In fact, PCA based faulty data detection algorithms and PCA based data aggregation algorithms have already been studied in the literature. The related works are summarized in Section 2.4.2.

In the previous literature, multivariate sensor data aggregation is performed at each node locally [29][31]. Considering the high complexity of PCA model and limited computational capacity of a sensor node, we propose a novel recursive-PCA (R-PCA) based multivariate fault-tolerant data aggregation algorithm. In the proposed algorithm, the multivariate sensor data training and aggregating operations are implemented based on clusters instead of local nodes. Besides, the R-PCA model recursively updates the transformation basis, which is more adaptable to the inner and outer changes of WSNs. The data fault detection accuracy, data restoration accuracy and network energy consumption are evaluated based on simulations. Simulation results show that the proposed algorithm improves the network performance on these aspects.

This chapter is organized as follows. The mathematical models are introduced in Section 5.2. Section 5.3 presents the details of the proposed fault-tolerant data aggregation algorithm. The performance evaluation of the proposed algorithm is demonstrated in Section 5.4, as compared to the conventional PCA models and algorithms. Finally, the content of this chapter is summarized in Section 5.5.

5.2 System Model

In this section, the conventional principal component analysis (PCA) model is briefly introduced and the network model used in this work is presented.

5.2.1 Conventional PCA

PCA is a statistical tool that is normally used to reduce the dimension of data [27]. The mathematical principle behind PCA is the change of basis. Given a matrix, its natural basis is an identity matrix, *i.e.*, $X = IX$, where X is the original data matrix consisted of m physical measurements and n observations. PCA aims at finding out a basis P, which makes the projection of X, *i.e.*, $Y = PX$, consist of less-dimensional and linearly uncorrelated principal components.

Since a symmetric matrix can be diagonalized by its orthogonal eigenvector matrix, the covariance matrix of X (defined in [27]) can be diagonalized as

$$C_X = \frac{1}{n-1} XX^T = E\Lambda E^T, \tag{5.1}$$

where the columns in E are the eigenvectors of C_X and Λ is a diagonal matrix with the eigenvalues of C_X. Based on (5.1), it can be derived that C_Y can be diagonalized by E^T as

$$
\begin{aligned}
C_Y &= \frac{1}{n-1} Y Y^T \\
&= E^T \frac{1}{n-1} X X^T (E^T)^T \\
&= E^T E \Lambda E^T E \\
&= \Lambda.
\end{aligned} \tag{5.2}
$$

Now, the matrix Y is linearly uncorrelated as the covariance matrix of Y is diagonal, but the dimension is still the same as X. The next step is to make Y less dimensional. The new dimension of Y is the number of principal components of X, labelled as l. It is decided by the cumulative percentage formula as

$$
\alpha = \frac{\sum_i^l \tilde{\Lambda}_i}{\sum_i^m \tilde{\Lambda}_i} \times 100\%, \tag{5.3}
$$

where $\tilde{\Lambda}$ is the reordered eigenvalue matrix Λ and α is determined by the specific application requirements, *e.g.*, 90%. Then the eigenvector matrix is reordered and reduced accordingly, $\tilde{E}_l$. Finally, the transformation basis (P) is set to the transposition of the reordered and reduced eigenvector matrix ($\tilde{E}_l$), *i.e.*, $P = \tilde{E}_l^T$. The generated matrix Y is an $l \times n$ linearly uncorrelated matrix, namely, $Y = PX = \tilde{E}_l^T X$.

5.2.2 Network Model

The proposed multivariate fault-tolerant data aggregation algorithm is based on the cluster tree topology. The network topology is demonstrated in Fig.5.1. As shown in Fig.5.1, the leaf sensor nodes send sensor readings to their cluster head. The cluster head is responsible for detecting and discarding the faulty data and aggregating the redundant data. Then the fault-free aggregated sensor data is forwarded to the sink. In the next section, the implementation of the data fault detection and data aggregation algorithm by exploitation of the cluster tree topology is explained in detail.

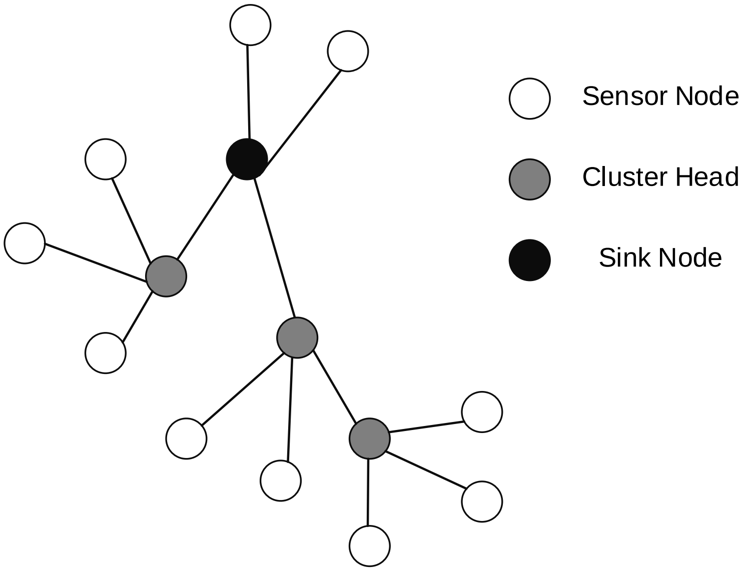


Figure 5.1: Cluster tree topology of wireless sensor networks.

5.3 R-PCA based Multivariate Fault-tolerant Data Aggregation Algorithm

5.3.1 Recursive-PCA based Fault Detection Method

The pseudocode of recursive-PCA (R-PCA) method is given in Algorithm 3. The specific operations in the method are depicted in the following paragraphs.

Standardization Given a raw data matrix X, $X = \left[\overrightarrow{x_1}^T, \overrightarrow{x_2}^T, \ldots, \overrightarrow{x_m}^T\right]^T$, where m is the number of physical measurements and $\overrightarrow{x_i}$ is the vector of observations, $\overrightarrow{x_i} = \{x_i(1), x_i(2), \ldots, x_i(t), \ldots\}$. It is standardized to a new zero-mean and unit-variance matrix ($\overline{X}$) first, in order to mitigate the influence of different units. It is mathematically presented as

$$\overline{x}_i(t) = \frac{x_i(t) - \mu_i(t)}{\sigma_i(t)}, \tag{5.4}$$

where $\mu_i(t)$ and $\sigma_i(t)$ are the mean value and standard variance of $\overrightarrow{x_i}$.

Since $\mu_i(t) = \frac{1}{t}\sum_{j=1}^{t} x_i(j)$, therefore the mean value can be recursively updated by

$$\mu_i(t+1) = (1-\beta) \cdot \mu_i(t) + \beta \cdot x_i(t+1), \tag{5.5}$$

Algorithm 3 R-PCA based Fault Detection Method

```
Initialization:
standardize $X \Rightarrow \overline{X} \sim N(0,1)$
calculate $E$ and $\Lambda$ of $\frac{1}{n-1}\overline{X}\,\overline{X}^T$
initialize $\mu_{SPE}$ and $\sigma_{SPE}$
Recursion:
update $\mu_x$ and $\sigma_x$ and standardize $X(t)$
rank $\Lambda$, $E$ and calculate the number of PCs, $l$
reduce $\tilde{E} \rightarrow \tilde{E}_l$ and $\tilde{\Lambda} \rightarrow \tilde{\Lambda}_l$
calculate $SPE(t)$
if $|SPE(t) - \mu_{SPE}| >= \xi \cdot \sigma_{SPE}$ then
    (fault detected)
else
    update $E$ and $\Lambda$
    update $\mu_{SPE}$ and $\sigma_{SPE}$
end if
```

where $\beta = \frac{1}{t+1}$ is the forgetting factor [63]. Similarly, the variance is recursively updated by

$$\sigma_i^2(t+1) = (1-\beta) \cdot \sigma_i^2(t) + \beta \cdot (x_i(t+1) - \mu_i(t+1))^2. \tag{5.6}$$

Principal Components (PCs) Rank the eigenvalues and reorder the eigenvector matrix accordingly. The number of principal components (PCs), l, can be calculated by cumulative percentage formula as

$$\eta = \frac{\sum_{i=1}^{l} \tilde{\Lambda}_i}{\sum_{i=1}^{m} \tilde{\Lambda}_i}, \tag{5.7}$$

where the value of l is determined by the application specified η, *e.g.*, 90%.

SPE Score SPE score is used as the detection criterion [28], which is calculated as

$$SPE(t) = ||\overline{X}(t) - \tilde{E}_l \tilde{E}_l^T \overline{X}(t)||^2. \tag{5.8}$$

Additionally, μ_{SPE} and σ_{SPE} are updated with (5.9) and (5.10),

$$\mu_{SPE}(t+1) = (1-\beta) \cdot \mu_{SPE}(t) + \beta \cdot SPE(t+1), \tag{5.9}$$

$$\sigma^2_{SPE}(t+1) = (1-\beta)\cdot\sigma^2_{SPE}(t) + \beta\cdot(SPE(t+1) - \mu_{SPE}(t+1))^2, \tag{5.10}$$

where β is the forgetting factor.

Eigenvectors and Eigenvalues Update The covariance matrix of $\overline{X}$ at time instance $t+1$ is

$$\begin{aligned} C_X(t+1) &= \frac{1}{t}\sum_{j=1}^{t+1}\overline{X}(j)\overline{X}^T(j) \\ &= (1-\varepsilon)C_X(t) + \varepsilon\cdot\overline{X}(t+1)\overline{X}^T(t+1) \\ &= C_X(t) + \varepsilon\cdot(\overline{X}(t+1)\overline{X}^T(t+1) - C_X(t)), \end{aligned} \tag{5.11}$$

where ε is the modifying factor and usually < 0.01.

The eigenvector decomposition of covariance matrix at time instance is given by

$$\begin{aligned} C_X(t+1) &= E(t+1)\Lambda(t+1)E^T(t+1) \\ &= E(t)\Lambda(t)E^T(t) + \varepsilon(\overline{X}(t+1)\overline{X}^T(t+1) - E(t)\Lambda(t)E^T(t)) \\ &= E(t)[(1-\varepsilon)\Lambda(t)]E^T(t) + \varepsilon\overline{X}(t+1)\overline{X}^T(t+1), \end{aligned} \tag{5.12}$$

let $A(t+1) = \overline{X}^T(t+1)E(t)$, then

$$C_X(t+1) = E(t)[(1-\varepsilon)\Lambda + \varepsilon A^T(t+1)A(t+1)]E^T(t). \tag{5.13}$$

It can be further decomposed as

$$(1-\varepsilon)\Lambda + \varepsilon A^T(t+1)A(t+1) = U(t+1)\Sigma(t+1)U^T(t+1). \tag{5.14}$$

Therefore, the eigenvectors and eigenvalues can be updated by

$$E(t+1) = E(t)U(t+1), \tag{5.15}$$

$$\Lambda(t+1) = \Sigma(t+1). \tag{5.16}$$

Let

$$U(t+1) = I + H_1(t+1), \tag{5.17}$$

$$\Sigma(t+1) = (1-\varepsilon)\Lambda(t) + H_2(t+1). \tag{5.18}$$

Based on the first-order perturbation theory [64], the H_1 and H_2 are derived as

$$H_1(t+1) = \begin{cases} 0 & , i = j, \\ \frac{\varepsilon A_i(t+1)A_j(t+1)}{(1-\varepsilon)(\Lambda_j^2(t)-\Lambda_i^2(t))+\varepsilon(A_j^2(t+1)-A_i^2(t+1))} & , i \neq j. \end{cases} \tag{5.19}$$

and

$$H_2(t+1) = \begin{cases} \varepsilon A_i^2 & , i = j, \\ 0 & , i \neq j. \end{cases} \tag{5.20}$$

Finally, E and Λ are updated by (5.15)-(5.20).

5.3.2 Multivariate Fault-tolerant Data Aggregation Algorithm

With the development of embedded system, one sensor node is embedded with multiple sensors. Correspondingly, the data matrix generated by one node consists of multiple variables, e.g. temperature, humidity and etc. Therefore, the current data processing in sensor network moves forward from univariate to multivariate data model.

Given a sensor node i, its data matrix (X_i) is mathematically presented as

$$X_i = \begin{bmatrix} x_{i,1}(1) & x_{i,1}(2) & \dots & x_{i,1}(T) \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,M}(1) & x_{i,M}(2) & \dots & x_{i,M}(T) \end{bmatrix}, \tag{5.21}$$

where M is the number of physical variables and T is the number of observations.

Based on the cluster tree network topology, the fault detection and data aggregation are implemented by the cluster head, CH. CH collects the multivariate sensor data from its members and reorganizes the data by their physical properties. For example, if there are N nodes in the cluster including CH, there will be M data matrices, each like

$$\hat{X}_m = \begin{bmatrix} x_{1,m}(1) & x_{1,m}(2) & \dots & x_{1,m}(T) \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,m}(1) & x_{N,m}(2) & \dots & x_{N,m}(T) \end{bmatrix}. \tag{5.22}$$

The R-PCA method is implemented based on each $\hat{X}_m$, in order to detect and discard the faulty data and further aggregate the redundant data. The network topology and data flow are shown in Fig.5.2 and the algorithm works as follows.

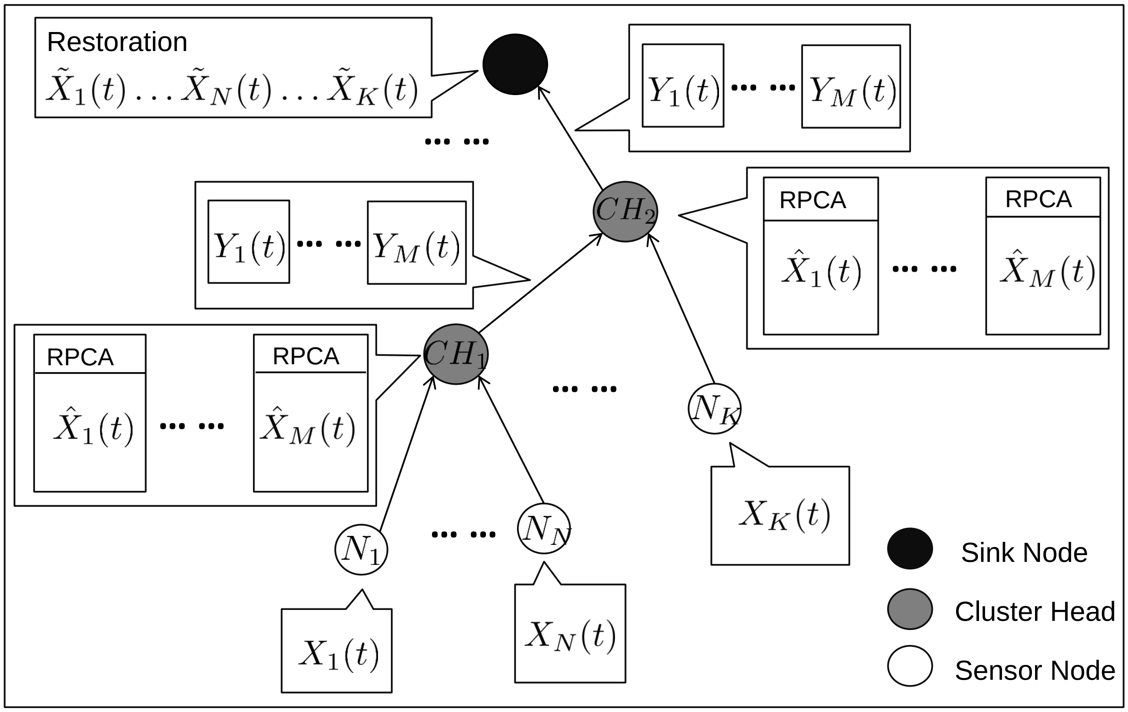

Restoration
$\tilde{X}_1(t) \dots \tilde{X}_N(t) \dots \tilde{X}_K(t)$
$Y_1(t)$ ··· ··· $Y_M(t)$
CH_2
RPCA
$\hat{X}_1(t)$ ··· ··· $\hat{X}_M(t)$
CH_1
N_K
$X_K(t)$
N_1
N_N
$X_1(t)$
$X_N(t)$
Sink Node
Cluster Head
Sensor Node

Figure 5.2: Multivariate fault-tolerant data aggregation algorithm.

1. The leaf sensor nodes (*e.g.*, N_1,N_N) send their data to lower level cluster head (CH_1).

2. CH_1 reorganizes the sensor readings from its descendants to $\{\hat{X}_1 \dots \hat{X}_M\}$. For each matrix, the data is diagnosed and aggregated by R-PCA method. CH_1 sends the principal components to a higher level cluster head.

3. At the higher level cluster head (CH_2), the local principal components (PCs) are calculated. Then CH_2 sends its own PCs and forwards the PCs from lower levels as well.

4. At the sink node, all the sensor readings are restored based on the received PCs.

5.4 Performance Evaluation

5.4.1 Univariate Scenario: Comparisons on Fault Detection

Three models are compared, PCA, exponentially weighted PCA (EW-PCA) and recursive PCA (R-PCA). The humidity sensor readings from Node 31 $\sim$ 40 are adopted as the test dataset [7]. The data fault occurs at random time instance and randomly chosen node with 10% probability. The faulty data varies from the original data with 5% offset. SPE score is adopted as the fault detection criterion. Detection accuracy and false alarm ratio with different thresholds are shown in Fig.5.3, where the detection accuracy is defined as

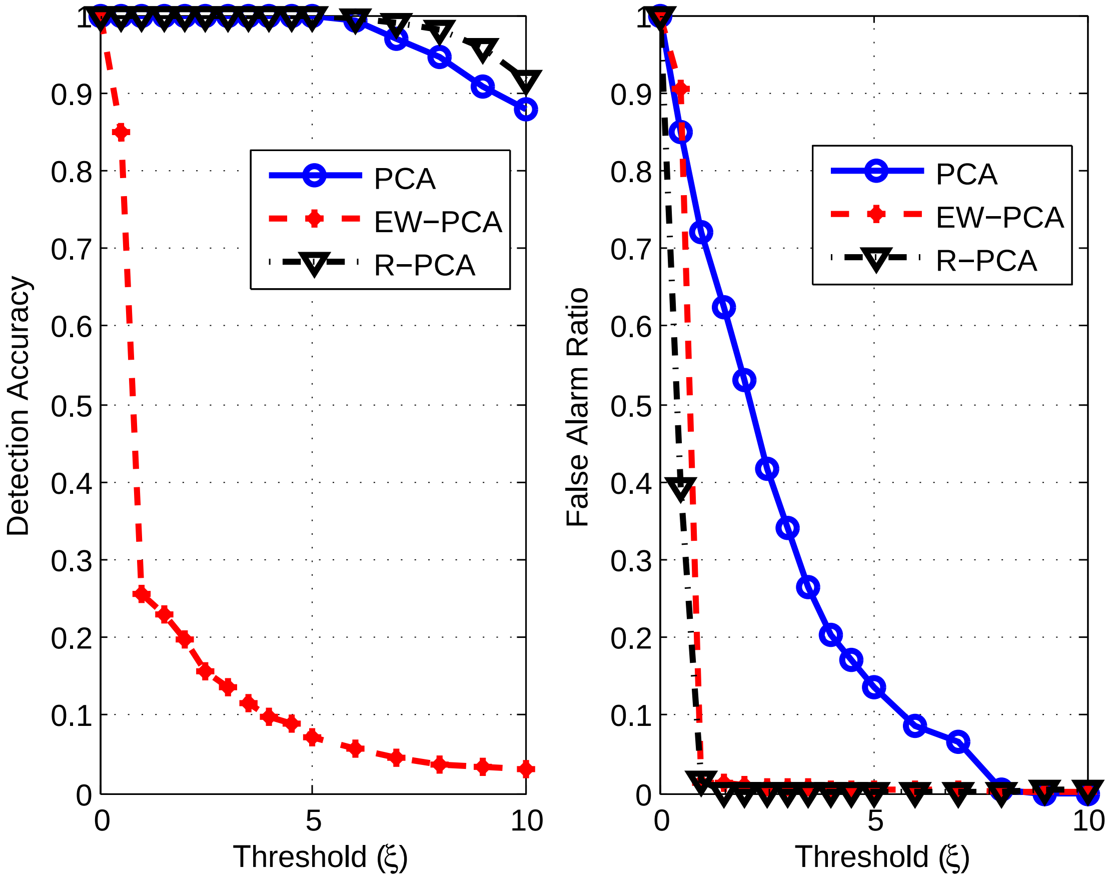


Figure 5.3: Detection accuracy and false alarm ratio of PCA, EW-PCA and R-PCA models with different thresholds (ξ).

$$da = \frac{TP}{TP + FN}, \tag{5.23}$$

and the false alarm ratio is given by

$$far = \frac{FP}{FP + TN}, \tag{5.24}$$

TP, FN, FP, TN are short for *true positive*, *false negative*, *false positive* and *true negative*, respectively [65]. In terms of the detection accuracy and false alarm ratio, the R-PCA model based fault detection algorithm outperforms the PCA and EW-PCA models (Fig.5.3). R-PCA based algorithm better adapts to the gradual changes of the network system, since the R-PCA model recursively updates the transformation basis.

5.4.2 Multivariate Scenario: Comparisons on Fault Detection

The proposed R-PCA (PR-PCA) based multivariate data aggregation algorithm is based on clusters while the conventional R-PCA (CR-PCA) based algorithm is implemented at local sensor nodes [31]. The fault detection accuracy of PR-PCA and CR-PCA algorithms is compared in this subsection. The generation of faulty data is similar to that in the univariate scenario. The difference is that not only humidity sensor data, but also temperature and voltage sensor readings are used.

5.4.2.1 Fault Probability

The faulty data in this simulation is randomly generated with a certain probability, termed as fault probability. The influence of the fault probability on the data fault detection is demonstrated in Fig.5.4. Fig.5.4 shows that the detection accuracy of PR-PCA is higher and the false alarm ratio of it is lower, as compared to CR-PCA. Additionally, with the increment in fault probability, the performance of CR-PCA decreases severely while PR-PCA is stable. It can be inferred that the PR-PCA is more trustworthy than CR-PCA when the probability of data fault occurrence is high.

5.4.2.2 Fault Offset

The faulty data is generated by adding an offset on the original data. Then the influence of the offset on the detection accuracy is investigated. The effect of offset on detection results tends to

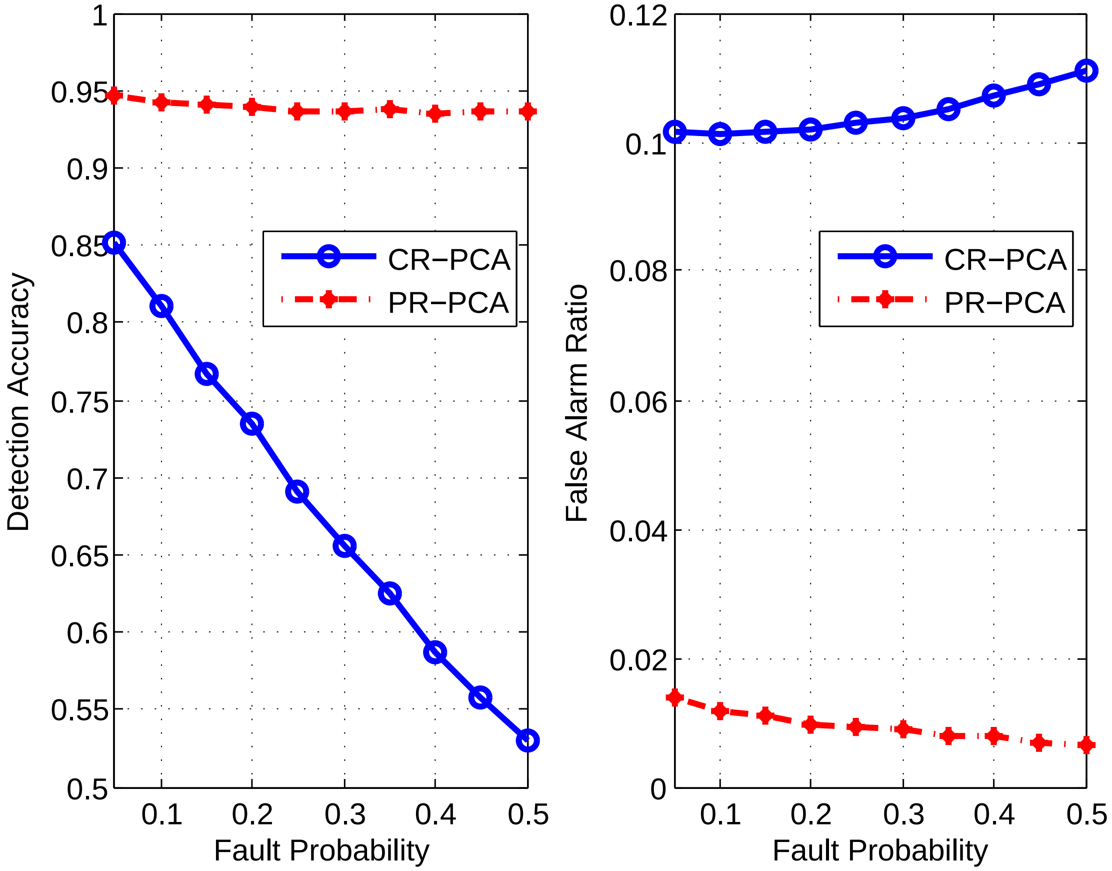


Figure 5.4: Effect of fault probability on data fault detection.

be stable when the fault offset is larger than 10%, as shown in Fig.5.5. It indicates that the fault detection accuracy is mainly affected by the threshold when the data outlier is over a certain value. Besides, PR-PCA outperforms CR-PCA whatever the fault offset is.

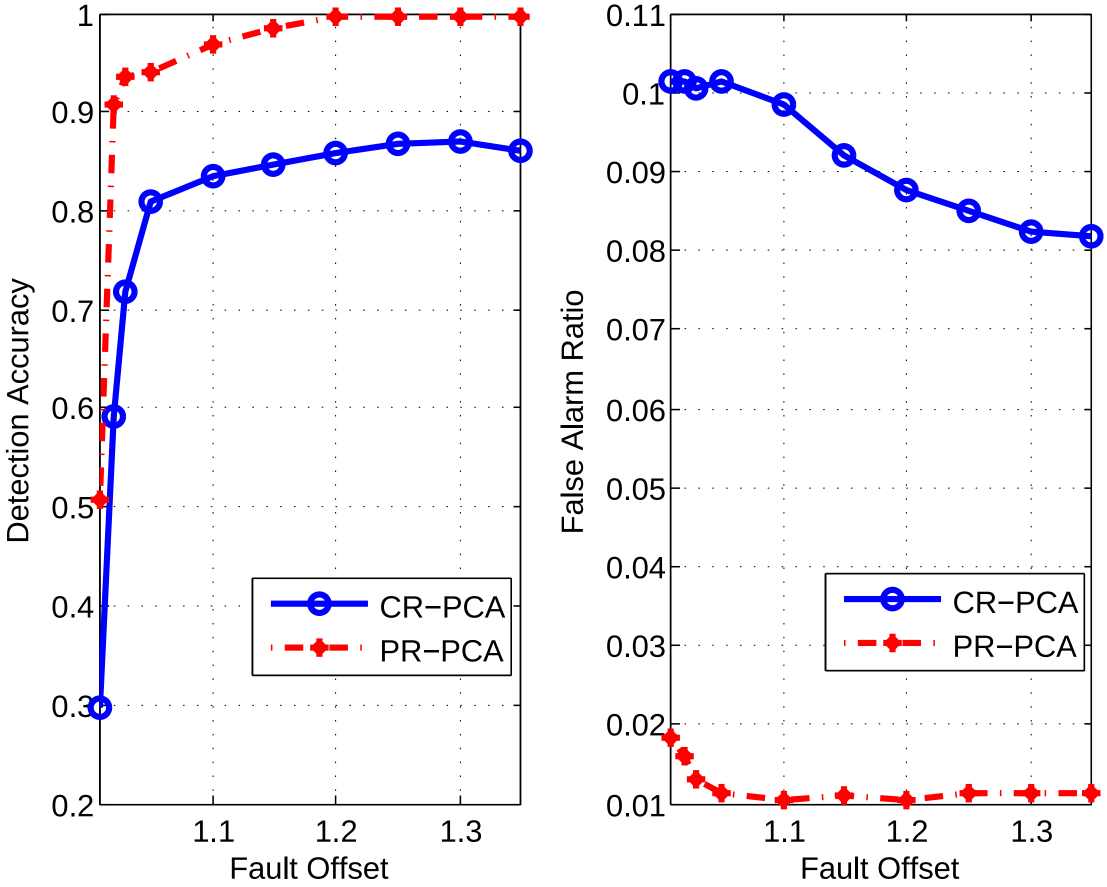


Figure 5.5: Effect of fault offset on data fault detection.

5.4.2.3 Threshold

Threshold ξ is used to evaluate whether the SPE score of certain data is beyond the normal range during the detection procedure. Fig.5.6 shows the effect of threshold on detection results.

From Fig.5.6, it can be seen that PR-PCA outperforms CR-PCA at both detection accuracy and false alarm ratio. Additionally, the detection accuracy decreases with the increasing threshold. The reason is that SPE scores of some faulty data are not dramatic enough to meet the higher detection thresholds. However, the false alarm ratio is stable after the threshold is

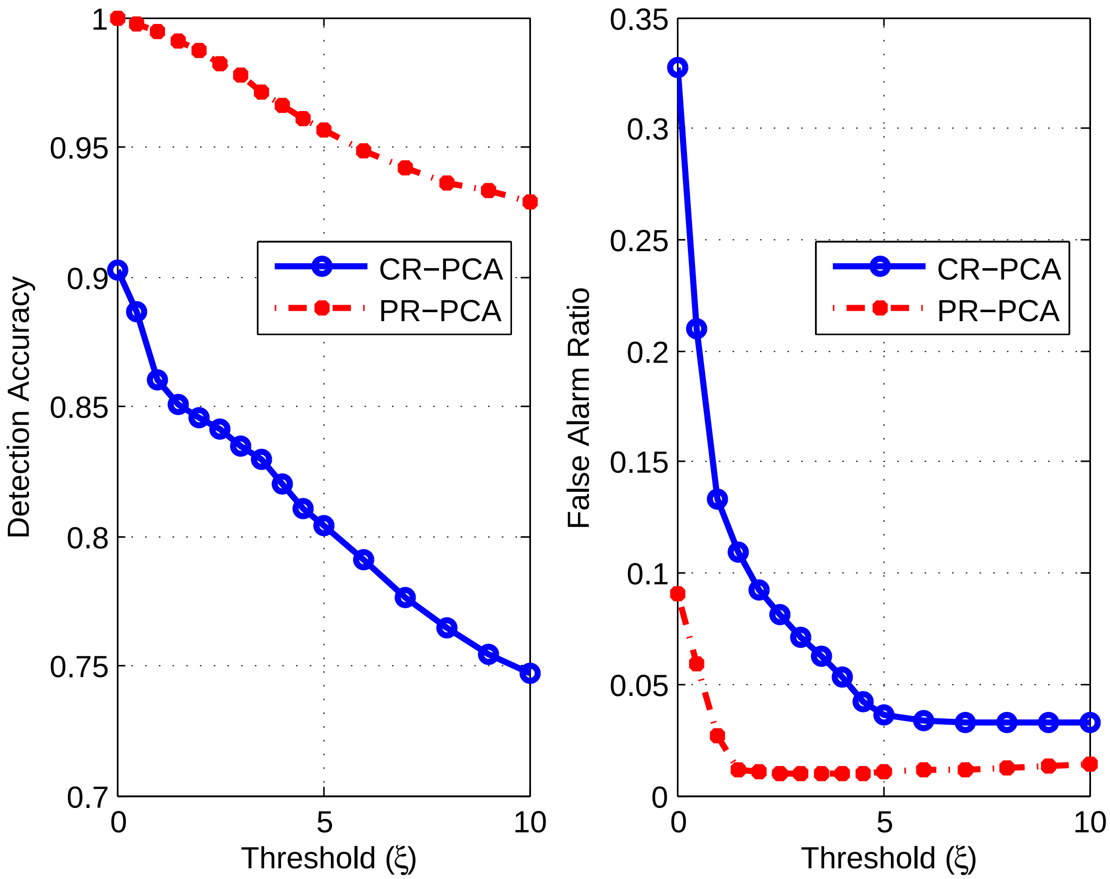


Figure 5.6: Effect of threshold ξ on the data fault detection.

larger than a certain value. This is because, the possibility of the normal data detected as faulty by mistake is lower when the threshold is large enough.

5.4.3 Multivariate Scenario: Comparisons on Restoration Error

The relative restoration error is a commonly used metric to evaluate the performance of data aggregation and restoration algorithms [29]. As mentioned in Section 5.3, the original data matrix X is transformed to a reduced dimensional matrix Y by $Y = \tilde{E}_l^T X$. At the receiver end, the data is recovered by $\tilde{X} = \tilde{E}_l Y$. Therefore, the relative restoration error is defined as

$$rre = \frac{||\tilde{X} - X||}{||X||}. \tag{5.25}$$

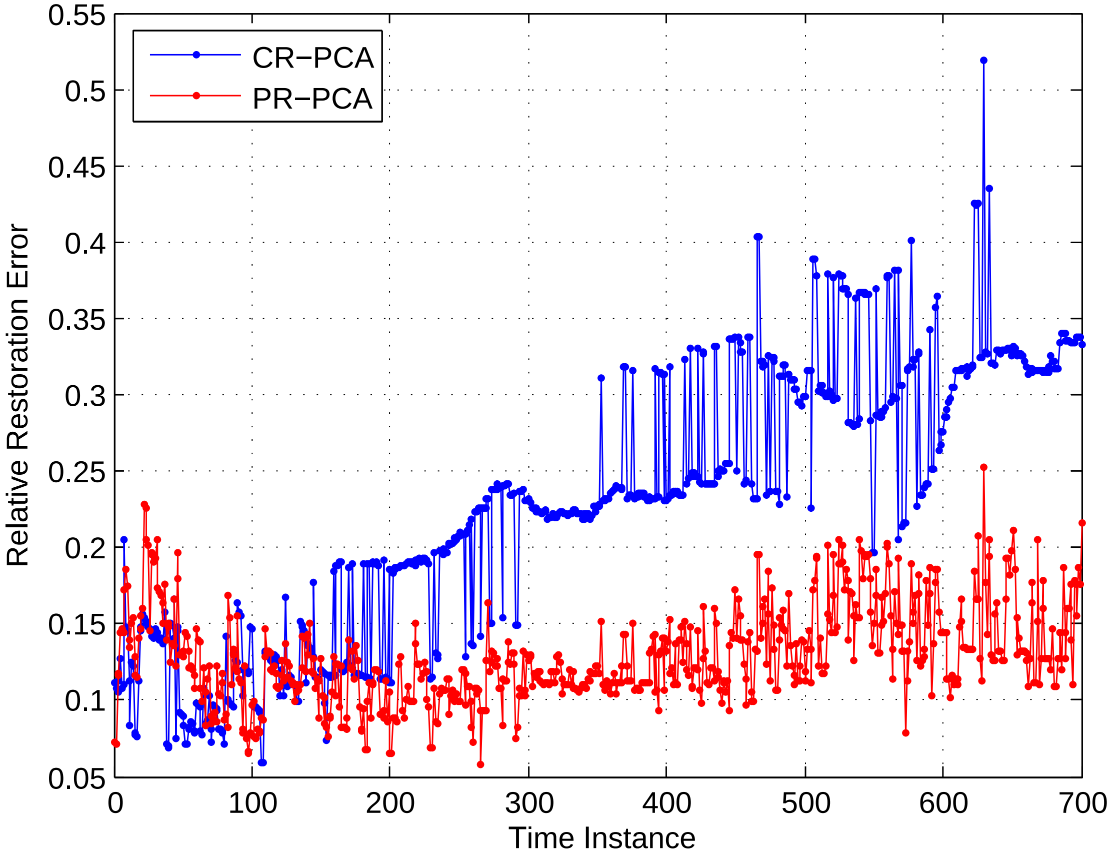


Figure 5.7: Relative reconstruction error of conventional and proposed algorithms, cluster size=10.

Fig.5.7 shows the relative restoration error of the conventional and proposed algorithms.

Cluster size refers to the number of sensor nodes in the cluster. More specific values of *rre* of the 700 time instances are summarized in Table 5.1.

Table 5.1: Relative Restoration Error

	rre_mean	*rre_std*	*rre_max*	*rre_min*
CR-PCA	0.2291	0.0879	0.5189	0.0587
PR-PCA	0.1289	0.0313	0.2524	0.0577

Both Fig.5.7 and Table 5.1 demonstrate that in terms of the restoration accuracy, the proposed cluster based multivariate data aggregation algorithm outperforms the conventional local one. This is because the correlation between sensor readings from neighbor nodes of the same physical parameter is stronger than that between sensor readings of different physical parameters from the same node.

5.4.4 Multivariate Scenario: Comparisons on Energy Cost

In order to evaluate the network energy consumption with different data aggregation algorithms, Micaz mote [66] is introduced as the energy consumption model in this work. The transmitting and receiving energy costs are 720 and 110 nJ/bit, respectively. The energy consumed by each CPU instruction is 4 nJ/instruction. Additionally, the packet header is 28 bits, the preamble overhead is 160 bits and each component occupies 28 bits of payload.

The cumulative network energy consumption and the energy consumed at each time instance by exploitation of our proposed R-PCA based data aggregation algorithm (PR-PCA) is shown in Fig.5.8, as compared to the network energy consumption with conventional R-PCA based data aggregation algorithm (CR-PCA) and without any data aggregation algorithm (None). From Fig.5.8, it can be seen that both CR-PCA and PR-PCA reduce the network energy consumption, as compared to the baseline protocol. Meanwhile, PR-PCA costs nearly 21% less energy than CR-PCA. The reason is that the data compression degree at the cluster head is larger than that at the local sensor node when the cluster size is 10. Therefore, fewer data transmissions decrease the network energy consumption.

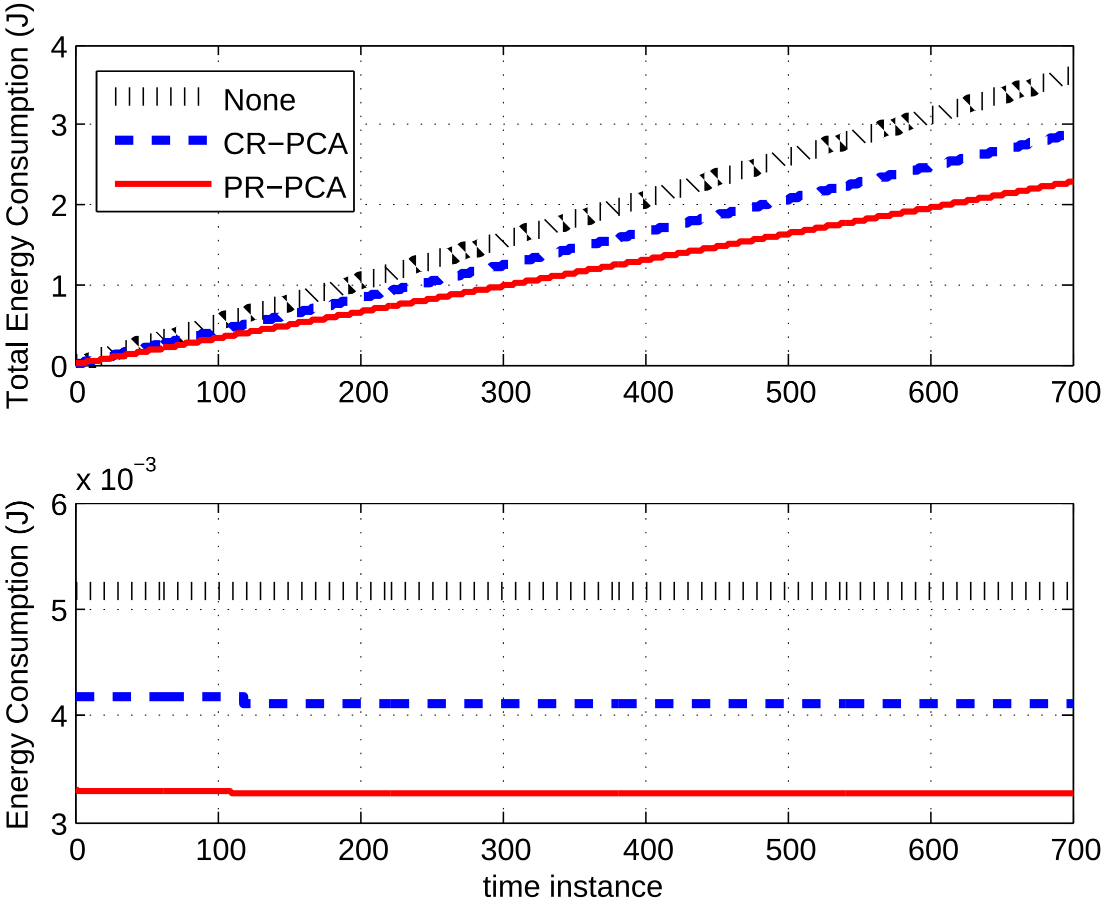


Figure 5.8: Network energy consumption of different algorithms, cluster size=10.

5.4.5 Discussion on Cluster Size

The simulation results in Subsection 5.4.3 and 5.4.4 are based on the cluster with ten nodes. To mitigate the limitation of the results, the effect of cluster size on the restoration error and network energy consumption is investigated below. Fig.5.9 shows the influence of cluster size on network energy consumption with different data aggregation algorithms.

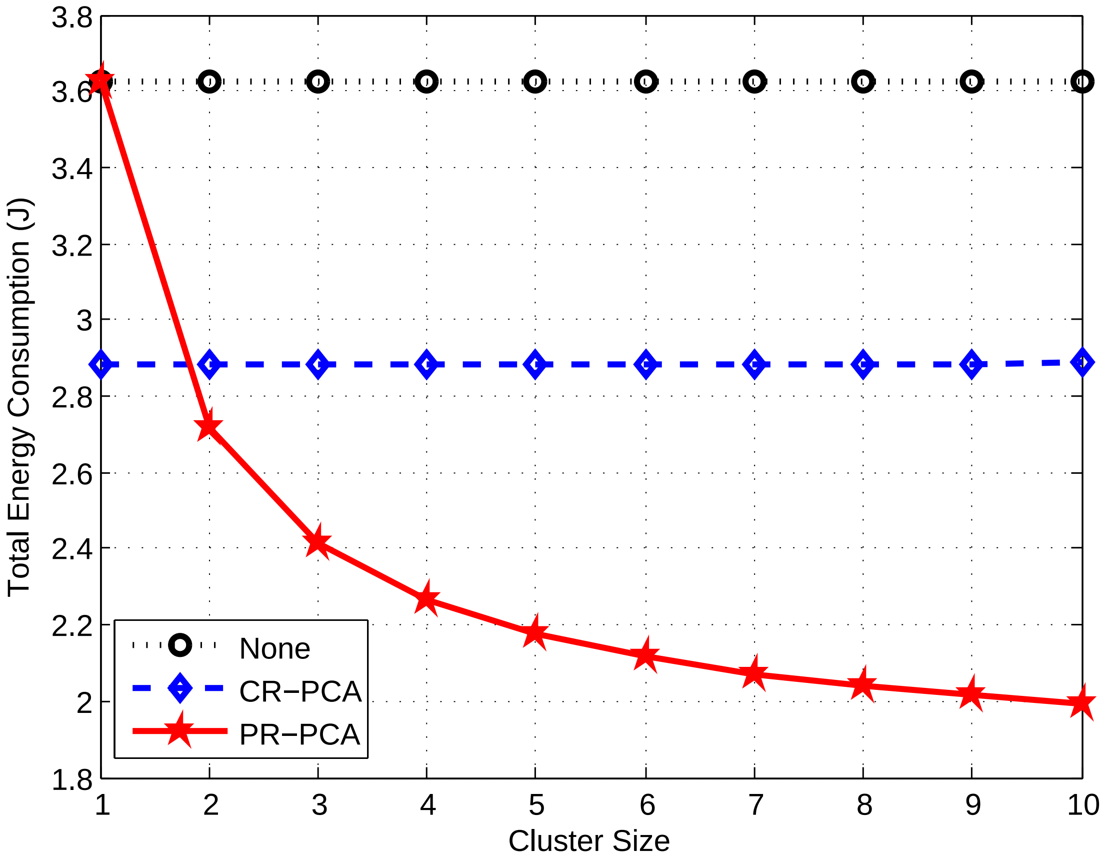


Figure 5.9: Network energy consumption of algorithms with different cluster sizes.

It can be seen that the network energy consumption of PR-PCA decreases with the increment in cluster size. The reason is that the sensor data can be further compressed when more sensor nodes within the same cluster. The network energy consumption is even higher than that of CR-PCA when the sensor nodes are not clustered.

The relative restoration error of algorithms with different cluster sizes is listed in Table 5.2. It can be seen that the *rre* of PR-PCA is smaller than CR-PCA. Besides, the relative restoration error of PR-PCA increases when the cluster contains more sensor nodes.

Table 5.2: Relative Restoration Error of CR-PCA and PR-PCA

Cluster Size	1	4	7	10
CR-PCA	0.0598	0.1022	0.1166	0.2291
PR-PCA	0	0.0866	0.0928	0.1289

Based on the discussions on Fig.5.9 and Table 5.1, it can be concluded that in terms of the cluster size, there is a trade-off between the network energy cost and the restoration accuracy. In practical scenarios, the specific cluster size is determined by the application requirements.

5.5 Chapter Summary

In this chapter, a new recursive principal component analysis (R-PCA) model is proposed, which recursively updates the transformation basis. As compared to the conventional PCA model and EW-PCA model, the R-PCA model better adapts to the changes of wireless sensor networks and the R-PCA based fault detection method improves the fault detection accuracy. The R-PCA model based multivariate fault detection and data aggregation algorithm considers the multiple physical parameters in WSNs and processes the data based on clusters. In comparison with the conventional local-based algorithm, the proposed algorithm decreases the restoration error and reduces the network energy consumption, because the correlation between sensor readings from neighbor nodes of the same physical parameter is stronger than that of different parameters from the same node.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, three major challenges in wireless sensor networks were investigated in detail, namely, energy efficiency, data fault and data redundancy. In order to overcome these problems in WSNs, three algorithms were proposed in Chapter 3-5, respectively.

In Chapter 3, a novel sensor scheduling mechanism was proposed, aiming at improving the network energy efficiency. The basic principle behind this mechanism was reducing the data transmission based on the highly spatial correlation of sensor data. More specifically, all the sensor nodes within the network were clustered by the proposed adaptive DK-means algorithm based on the spatial correlation of sensor data first. Then the order and duration of sensor nodes working as cluster representatives were determined by the new sensor scheduling algorithm. Instead of all the sensor nodes within the network, only the cluster representative nodes generated and transmitted sensor data at the meantime, so that the energy costed by sensing and transmitting were saved. Simulations conducted in OPNET proved that the proposed sensor scheduling algorithm reduced the energy cost, as compared to the baseline ZigBee protocol.

In order to detect the faulty data, a novel distributed fault detection algorithm based on temporal and spatial correlation of sensor data was proposed in Chapter 4. Since the physical parameters changed continuously in nature, the normal range of the sensor measurements to be collected could be predicted by both its own historical observations and its neighbor sensor readings. Therefore, the abnormal sensor data could be detected by the dramatical variance from the normal range. Besides, both the result of temporal detection and received signal strength indicator were used as weights in the spatial detection procedure, which further improved the detection accuracy. Simulations based on both practical and synthetic datasets showed that the proposed algorithm improved the detection accuracy indeed, as compared to the distributed fault detection algorithms in the literature.

The last contribution of this thesis was the reduction of the data redundancy. In Chapter 5, a recursive principal component analysis (R-PCA) based data aggregation algorithm was proposed. At the beginning, the R-PCA model was introduced based on the modification of basic PCA model so that the transformation basis could be recursively updated. Based on the analysis of data correlation, the proposed data aggregation algorithm was implemented along the cluster tree instead of the local nodes, since the correlation between the same physical measurements from neighbor nodes was stronger than that of different physical measurements from the same node. For this reason, sensor readings from leaf nodes were aggregated by R-PCA model at cluster head before being forwarded to the sink node. As compared to the conventional local data aggregation algorithm, the proposed algorithm improved the restoration accuracy and reduced the network energy consumption.

In summary, this thesis proposed several solutions to the urgent challenges in WSNs so that the network performance could be improved. With more efforts on performance enhancement in WSNs, the networks will be ubiquitously used soon.

6.2 Future Work

In the future, some aspects of the proposed algorithms are still worthwhile to be further investigated. Some potential research works are summarized as follows.

- Both the sensor scheduling algorithm in Chapter 3 and data aggregation algorithm in Chapter 5 were cluster-based. As discussed in the performance evaluation sections, energy consumption could be further improved by larger cluster size. However, over-sized cluster could deteriorate data reliability. Therefore, it is considerable to balance the energy consumption and data reliability during the procedure of cluster size determination. According to the adaptive DK-means clustering algorithm, the number of clusters were adaptively decided by the two thresholds. How to optimally select the thresholds could be further investigated, in order to determine the optimal cluster size.

- Timing of the algorithms was evaluated by the simulation time in this thesis. In the future, different time complexity analysis methods will be used to evaluate the algorithms. In terms of the simulation time, the distributed fault detection algorithm required more time than other algorithms in literature, although it outstandingly improved the detection accuracy. A new study could be started on reducing the simulation time of the algorithm.

- All the proposed algorithms in this thesis were simulated in either OPNET or MATLAB. In other words, all the algorithm performance evaluations were based on the simulated experiments. Since applicability is important in WSNs, practical experiments should be conducted to evaluate the algorithms in the future.

Bibliography

[1] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[2] Chris Masden. The paradigm shift of the internet of things and its impact on and in the mems and sensors industry. In *MEMS Executive Congress*. MEMS Industry Group, 2014.

[3] SIG Bluetooth. Bluetooth specification, 2010.

[4] ZigBee Alliance. Ieee 802.15. 4, zigbee standard. *On http://www. zigbee. org*, 2009.

[5] LAN/MAN Standards Committee et al. Part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). *IEEE Computer Society*, 2006.

[6] Bernard Rosner. *Fundamentals of biostatistics*. Cengage Learning, 2010.

[7] Peter Bodik, Wei Hong, Carlos Guestrin, Sam Madden, Mark Paskin, and Romain Thibaux. Intel lab data. *Online dataset*, 2004.

[8] Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.

[9] Hongbo Jiang, Shudong Jin, and Chonggang Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):1064–1071, 2011.

[10] Liu Xiang, Jun Luo, and Catherine Rosenberg. Compressed data aggregation: Energy-efficient and high-fidelity data collection. *Networking, IEEE/ACM Transactions on*, 21(6):1722–1735, 2013.

[11] Zhe Chen, Juri Ranieri, Rongting Zhang, and Martin Vetterli. Dass: Distributed adaptive sparse sensing. 2013.

[12] Y-Y Zhang, H-C Chao, Min Chen, Lei Shu, C-H Park, and M-S Park. Outlier detection and countermeasure for hierarchical wireless sensor networks. *IET information security*, 4(4):361–373, 2010.

[13] Colin O'Reilly, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. Anomaly detection in wireless sensor networks in a non-stationary environment. *Communications Surveys & Tutorials, IEEE*, 16(3):1413–1432, 2014.

[14] Miao Xie, Jiankun Hu, and Song Guo. Segment-based anomaly detection with approximated sample covariance matrix in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 26(2):574–583, 2015.

[15] Miao Xie, Song Han, Biming Tian, and Sazia Parvin. Anomaly detection in wireless sensor networks: A survey. *Journal of Network and Computer Applications*, 34(4):1302–1325, 2011.

[16] Sutharshan Rajasegarar, James C Bezdek, Christopher Leckie, and Marimuthu Palaniswami. Elliptical anomalies in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 6(1):7, 2009.

[17] Yang Zhang. *Observing the unobservable: distributed online outlier detection in wireless sensor networks*. University of Twente, 2010.

[18] Jun Fang and Hongbin Li. Optimal/near-optimal dimensionality reduction for distributed estimation in homogeneous and certain inhomogeneous scenarios. *Signal Processing, IEEE Transactions on*, 58(8):4339–4353, 2010.

[19] Bo Gong, Peng Cheng, Zhe Chen, Nian Liu, Liangqi Gui, and Frank de Hoog. Spatiotemporal compressive network coding for energy-efficient distributed data storage in wireless sensor networks. *Communication Letters*, 19(5):803–806, 2015.

[20] Hailong Li, Vinayaka Pandit, Andrew Knox, and Dharma P Agrawal. A novel characteristic correlation approach for aggregating data in wireless sensor networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–9. IEEE, 2013.

[21] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[22] Leonardo N Ferreira, AR Pinto, and Liang Zhao. Qk-means: a clustering technique based on community detection and k-means for deployment of cluster head nodes. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–7. IEEE, 2012.

[23] Fady Medhat, Rabie Ramadan, Ihab Talkhan, et al. Smart clustering for multimodal wsns. In *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2012 Seventh International Conference on*, pages 367–372. IEEE, 2012.

[24] Djamila Mechta, Saad Harous, Ismahane Alem, and Dounya Khebbab. Leach-ckm: Low energy adaptive clustering hierarchy protocol with k-means and mte. In *Innovations in Information Technology (INNOVATIONS), 2014 10th International Conference on*, pages 99–103. IEEE, 2014.

[25] Pat-Yam Tsoi, Chi-Tsun Cheng, and Nuwan Ganganath. A k-means-based formation algorithm for the delay-aware data collection network structure. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2014 International Conference on*, pages 384–388. IEEE, 2014.

[26] Hassan Harb, Abdallah Makhoul, David Laiymani, Ali Jaber, and Rami Tawil. K-means based clustering approach for data aggregation in periodic sensor networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on*, pages 434–441. IEEE, 2014.

[27] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[28] Shing-Chow Chan, HC Wu, and Kai Man Tsui. Robust recursive eigendecomposition and subspace-based algorithms with application to fault detection in wireless sensor networks. *Instrumentation and Measurement, IEEE Transactions on*, 61(6):1703–1718, 2012.

[29] Amirmohammad Rooshenas, Hamid R Rabiee, Ali Movaghar, and M Yousof Naderi. Reducing the data transmission in wireless sensor networks using the principal component analysis. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on*, pages 133–138. IEEE, 2010.

[30] Andre LL Aquino, Orlando S Junior, Alejandro C Frery, E Lins de Albuquerque, and Raquel AF Mini. Musa: multivariate sampling algorithmfor wireless sensor networks. *Computers, IEEE Transactions on*, 63(4):968–978, 2014.

[31] Christos Anagnostopoulos, Stathes Hadjiefthymiades, and Panagiotis Georgas. Pc3: principal component-based context compression: improving energy efficiency in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 72(2):155–170, 2012.

[32] Christos Anagnostopoulos and Stathes Hadjiefthymiades. Advanced principal component-based compression schemes for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 11(1):7, 2014.

[33] Vassilis Chatzigiannakis and Symeon Papavassiliou. Diagnosing anomalies and identifying faulty nodes in sensor networks. *Sensors Journal, IEEE*, 7(5):637–645, 2007.

[34] Rui Zhang, Ping Ji, Dinkar Mylaraswamy, Mani Srivastava, and Sadaf Zahedi. Cooperative sensor anomaly detection using global information. *Tsinghua Science and Technology*, 18(3):209–219, 2013.

[35] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

[36] Mohamed Younis, Izzet F Senturk, Kemal Akkaya, Sookyoung Lee, and Fatih Senel. Topology management techniques for tolerating node failures in wireless sensor networks: A survey. *Computer Networks*, 58:254–283, 2014.

[37] Bruno Bougard, Francky Catthoor, Denis C Daly, Anantha Chandrakasan, and Wim Dehaene. Energy efficiency of the ieee 802.15. 4 standard in dense wireless microsensor networks: Modeling and improvement perspectives. In *Design, Automation, and Test in Europe*, pages 221–234, 2008.

[38] Mehmet C Vuran, Özgür B Akan, and Ian F Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks*, 45(3):245–259, 2004.

[39] A Rajeswari and PT Kalaivaani. Energy efficient routing protocol for wireless sensor networks using spatial correlation based medium access control protocol compared with ieee 802.11. In *Process Automation, Control and Computing (PACC), 2011 International Conference on*, pages 1–6, 2011.

[40] Yajie Ma, Yike Guo, Xiangchuan Tian, and Moustafa Ghanem. Distributed clustering-based aggregation algorithm for spatial correlated sensor networks. *Sensors Journal, IEEE*, 11(3):641–648, 2011.

[41] Fei Yuan, Yiju Zhan, and Yonghua Wang. Data density correlation degree clustering method for data aggregation in wsn. *Sensors Journal, IEEE*, 14(4):1089–1098, 2014.

[42] Leandro A Villas, Azzedine Boukerche, Daniel Guidoni, Regina B Araujo, and Antonio Alfredo Ferreira Loureiro. An energy-aware spatial correlation mechanism to perform efficient data collection in wsns. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 882–889, 2011.

[43] Jialin Zhang and Yu Hen Hu. Data centric multi-shift sensor scheduling for wireless sensor networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 4594–4597, 2013.

[44] Dieter J Cichon and Thomas Kürner. Digital mobile radio towards future generation systems: Cost 231 final report. *COST European Cooperation in the Field of Scientific and Technical Research-Action*, 231, 1993.

[45] Claude Oestges and Arogyaswami J Paulraj. Propagation into buildings for broad-band wireless access. *IEEE Transactions on Vehicular Technology*, 53(2):521–526, 2004.

[46] Petr Jurčík and Anis Koubâa. The ieee 802.15. 4 opnet simulation model: reference guide v2. 0. 2007.

[47] Peizhong Yi, Abiodun Iwayemi, and Chi Zhou. Developing zigbee deployment guideline under wifi interference for smart grid applications. *Smart Grid, IEEE Transactions on*, 2(1):110–120, 2011.

[48] Sanjiv K Bhatia. Adaptive k-means clustering. In *FLAIRS Conference*, pages 695–699, 2004.

[49] Shuo Guo, Heng Zhang, Ziguo Zhong, Jiming Chen, Qing Cao, and Tian He. Detecting faulty nodes with data errors for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(3):40, 2014.

[50] Jinran Chen, Shubha Kher, and Arun Somani. Distributed fault detection of wireless sensor networks. In *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, pages 65–72. ACM, 2006.

[51] Peng Jiang. A new method for node fault detection in wireless sensor networks. *Sensors*, 9(2):1282–1294, 2009.

[52] Marwa Saihi, Boumedyen Boussaid, Ahmed Zouinkhi, and M Naceur Abdelkrim. Decentralized fault detection in wireless sensor network based on function error. In *Systems, Signals & Devices (SSD), 2013 10th International Multi-Conference on*, pages 1–5. IEEE, 2013.

[53] Meng Shuai, Kunqing Xie, Guanhua Chen, Xiuli Ma, and Guojie Song. A kalman filter based approach for outlier detection in sensor networks. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 4, pages 154–157. IEEE, 2008.

[54] Sai Ji, Yuan Shen-fang, Ting-huai Ma, and Chang Tan. Distributed fault detection for wireless sensor based on weighted average. In *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, volume 1, pages 57–60. IEEE, 2010.

[55] Sung Yul Lim and Yoon-Hwa Choi. Malicious node detection using a dual threshold in wireless sensor networks. *Journal of Sensor and Actuator Networks*, 2(1):70–84, 2013.

[56] Yang Zhang, Nicholas AS Hamm, Nirvana Meratnia, Alfred Stein, M van de Voort, and Paul JM Havinga. Statistics-based outlier detection for wireless sensor networks. *International Journal of Geographical Information Science*, 26(8):1373–1392, 2012.

[57] Tuan Anh Nguyen, Doina Bucur, Marco Aiello, and Kenji Tei. Applying time series analysis and neighbourhood voting in a decentralised approach for fault detection and classification in wsns. In *Proceedings of the Fourth Symposium on Information and Communication Technology*, pages 234–241. ACM, 2013.

[58] Kevin Ni, Nithya Ramanathan, Mohamed Nabil Hajj Chehade, Laura Balzano, Sheela Nair, Sadaf Zahedi, Eddie Kohler, Greg Pottie, Mark Hansen, and Mani Srivastava. Sensor network data fault types. *ACM Transactions on Sensor Networks (TOSN)*, 5(25), 2009.

[59] Sensorscope project. *http://sensorscope.epfl.ch./*.

[60] Huseyin Ugur Yildiz, Sinan Kurt, and Bulent Tavli. The impact of near-ground path loss modeling on wireless sensor network lifetime. In *Military Communications Conference (MILCOM), 2014 IEEE*, pages 1114–1119. IEEE, 2014.

[61] Guiyi Wei, Yun Ling, Binfeng Guo, Bin Xiao, and Athanasios V Vasilakos. Prediction-based data aggregation in wireless sensor networks: combining grey model and kalman filter. *Computer Communications*, 34(6):793–802, 2011.

[62] Friedrich Pukelsheim. The three sigma rule. *The American Statistician*, 48(2):88–91, 1994.

[63] Yingxin Xie, Xiangguang Chen, and Jun Zhao. Adaptive and online fault detection using rpca algorithm in wireless sensor network nodes. In *Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on*, pages 1371–1374. IEEE, 2012.

[64] J Martínez-Carranza, Francisco Soto-Eguibar, and Héctor Moya-Cessa. Alternative analysis to perturbation theory in quantum mechanics. *The European Physical Journal D*, 66(1):1–6, 2012.

[65] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[66] Christos Anagnostopoulos and Stathes Hadjiefthymiades. Context compression: using principal component analysis for efficient wireless communications. In *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, volume 1, pages 206–215. IEEE, 2011.

Appendix A

Datasets

The research in this thesis is based on the intrinsic data correlation character of sensor data. Therefore, a large amount of sensor data has been used. The resources of these sensor data are the public datasets published by Intel Berkeley Research Lab [7] and SensorScope Group [59]. Brief introductions to these two labs are given below.

A.1 Intel Berkeley Research Lab

54 Mica2Dot sensor nodes were deployed at Intel Berkeley research lab from 2/28/2004-4/5/2004. The sensor nodes were settled in the lab according to Fig.A.1.

Each sensor node collected four physical parameters: temperature, humidity, light and voltage. Specifically, temperature was in Celsius degree. Humidity ranged from 0-100%, which was temperature corrected relative humidity. Light was in Lux and voltage was in volts. The data was sampled every 30 seconds. 2.3 million sensor readings were collected in total.

A.2 SensorScope Station at Grand-St-Bernard

Funded by NCCR MICS, several SensorScope stations were founded in Switzerland to monitor and collect the environmental parameters. The data used in our research was from the SensorScope station located at Grand-St-Bernard pass between Switzerland and Italy, which was collected in September 2007. The sensor nodes were practically deployed as Fig.A.2.

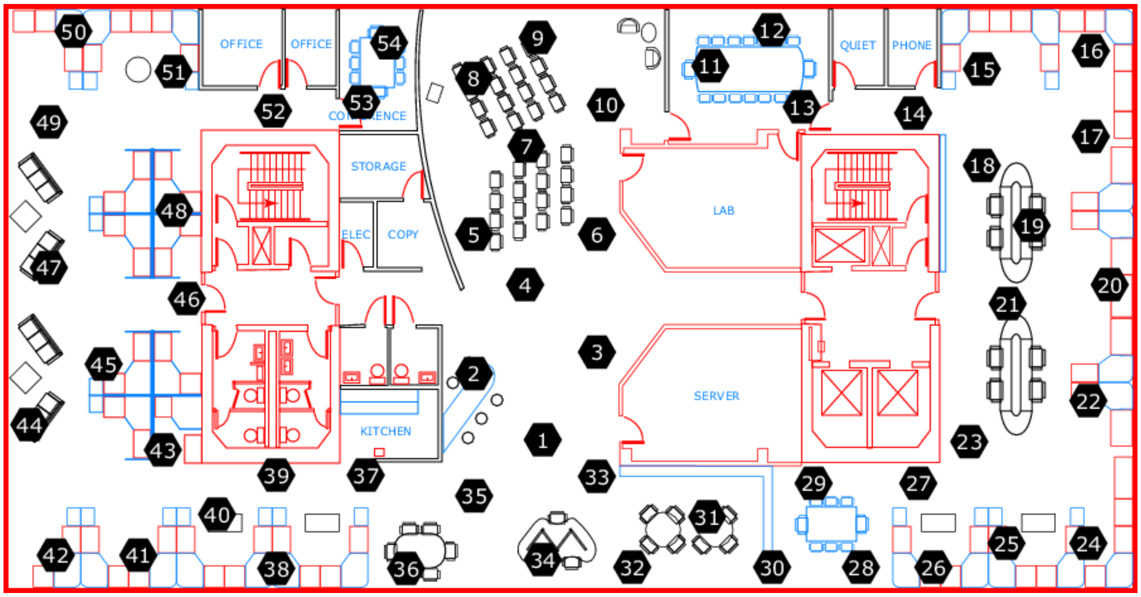


Figure A.1: Sensor nodes deployment at Intel Berkeley research lab.

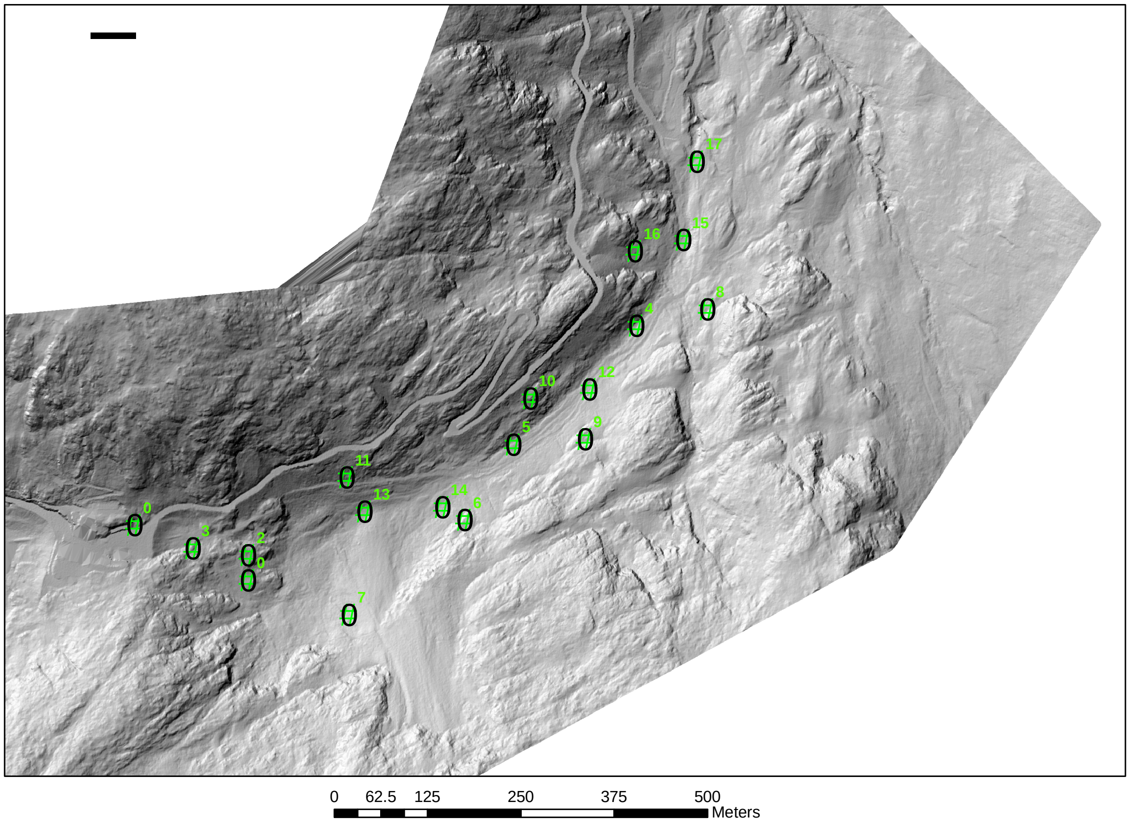


Figure A.2: Sensor nodes deployment at Grand-St-Bernard SensorScope stations.

Each sensor node collected several environmental parameters, *i.e.*, temperature, relative humidity, solar radiation, soil moisture, watermark, rain meter, wind speed and wind direction. Besides, the node collected the parameters of itself as well, *e.g.*, voltage and current. The sensor nodes transmitted data back to the base station every 2 minutes.

Curriculum Vitae

Name:	Tianqi Yu
Post-Secondary Education and Degrees:	2013 - present, M.E.Sc Electrical and Computer Engineering The University of Western Ontario London, Ontario, Canada 2009 - 2013, B.Eng (Hons) Communication Engineering Wuhan University Wuhan, Hubei, China
Honours and Awards:	NSERC CREATE Scholar, 2013-2015 Best Student Paper Award, IEEE VTC 2015 Spring
Related Work Experience:	Teaching Assistant The University of Western Ontario 2013 - 2015 Research Assistant The University of Western Ontario 2013-2015

Publications:

[1] T.Yu, A.Akhtar, A.Shami and X.Wang, ”Energy-Efficient Scheduling Mechanism for Indoor Wireless Sensor Networks,” in *Proc. IEEE VTC Spring,* May 2015.
[2] T.Yu, A.Akhtar, X.Wang and A.Shami, ”Temporal and Spatial Correlation based Distributed Fault Detection in Wireless Sensor Networks,” in *Proc. IEEE CCECE,* May 2015.