ENERGY EFFICIENT SCHEMES IN A MOBILE SENSOR NETWORK WITH APPLICATION IN AUTOMATED MONITORING OF WATER QUALITY

by

Tongxin Shu

B.Eng., Xiamen University, 2014

M.Sc., The University of British Columbia, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

January 2020

© Tongxin Shu, 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the dissertation entitled:

Energy Efficient Schemes in A Mobile Sensor Network With Application in Automated Monitoring of Water Quality

submitted by	Tongxin Shu	in partial fulfillment of the requirements for
the degree of	Doctor of Philosophy	
in	Electrical and Computer Enginee	ring
Examining Co	mmittee: Electrical and Computer Engineerir	φ
Co-supervisor	(Pro-tem)	6
Clarence W. d	e Silva, Mechanical Engineering	
Co-supervisor		
Ryozo Nagam	une, Mechanical Engineering	
Supervisory C	ommittee Member	
Alan Mackwor	rth, Computer Science	
University Exa	aminer	
José R. Martí	aminer	
University Exc		

Abstract

The present thesis addresses the power management issues in a mobile sensor network, with application in automated water quality monitoring. A water quality monitoring platform typically involves a wireless sensor network (WSN), in which a number of mobile sensor nodes (SN) are deployed in the water body to constantly collect the water-related sensory data such as the dissolved oxygen, pH value, temperature, oxidation-reduction potential, and electrical conductivity. This data is used to compute water quality index values, transmit them via some routing schemes, and eventually make them accessible to the water quality professionals, governing agencies, or the public.

Power management is nontrivial in the monitoring of a remote environment, especially when longterm monitoring is anticipated. However, constrained by the limited energy supply and internal characteristics of the devices, without proper power management, the devices may become nonfunctional within the networked monitoring system, and as a consequence, the data or events captured during the monitoring process will become inaccurate or non-transmittable. Research is proposed here to develop three distinct approaches for energy conservation in a sensor network, and apply them for automated monitoring of the quality of water in an extensive and remote aquatic body.

This thesis analytically develops and applies several energy efficient schemes for power management in the automated spatiotemporal monitoring of the quality of water in an extensive and remote aquatic environment. In general, the schemes for power management of a sensor network can be investigated from a number of aspects and schemes. Those schemes typically range from physical layer optimization to network layer solutions. Meanwhile, depending on the specific

applications, some energy efficient methodologies are custom-designed, and thus have limitations when used in other applications. Given this background, three energy efficient methods are proposed in this thesis for conserving energy within a WSN. Those proposed three methods, including DDASA, Hybrid DPS and GCVD, are studied on both the sensor node level and the system level, which energy-efficiently reduce the energy consumption and save extra energy thereby prolonging the life of the WSN. It is expected that the proposed methods will be applicable in other spatiotemporal monitoring applications.

Lay Summary

This thesis contains the research that mainly aims to address the power management issues of a Wireless Sensor Network (WSN) and proposes energy efficient schemes for environmental monitoring applications where a WSN is utilized. The developed schemes can be considered as schemes designed on the sensor node level, node-to-node communication level, and in geographical deployment of the sensor nodes. It is envisaged that, with the developed schemes, the lifetime of the WSN is effectively prolonged while the accuracy of the collected measurements is maintained at a desired high level.

Preface

The research presented in this thesis has been conducted under the guidance and supervision of Dr. Clarence W. de Silva of the Mechanical Engineering department at the University of British Columbia. Dr. de Silva provided and supervised the overall water quality monitoring project and acquired resources and funding for the project. Dr. de Silva also suggested the research topic and methodologies for addressing the practical issues of the project. Dr. Bhargava provided some funding for the candidate. The co-supervisor pro-tem is Dr. Lutz Lampe.

Chapter 2 is based on the publication [Tongxin Shu, Min Xia, Jiachong Chen, and Clarence W. de Silva, "An Energy Efficient Adaptive Sampling Algorithm in A Sensor Network for Automated Water Quality Monitoring," *Sensors* 17.11 (2017):2511]. Tongxin Shu was responsible for all the major designed algorithms and proposed concepts for this research. Min Xia contributed analysis tools while Jiahong Chen performed the experimental support. Clarence W. de Silva was the supervisory author and involved throughout the research and manuscript composition.

Chapter 3 is based on a paper published in *IEEE Internet of Things Journal* [Tongxin Shu, Jiahong Chen, Vijay Bhargava, Clarence W. de Silva, "An Energy-Efficient Dual Prediction Scheme Using LMS filter and LSTM in Wireless Sensor Networks for Environment Monitoring"]. Tongxin Shu was responsible for the designed algorithm and proposed concepts. Jiahong Chen participated in the coding work for implementing the proposed algorithm and conducting the experiment. Clarence W. de Silva was the supervisory author and involved throughout the research and manuscript composition. Vijay Bhargava provide some general guidance.

Chapter 4 is based on a paper that is going to be resubmitted [Tongxin Shu, Jiahong Chen, Vijay Bharagava, Clarence W. de Silva, "Energy-efficient and connectivity-guaranteed coverage optimization for mobile wireless sensor networks"]. Tongxin Shu was responsible for the research concepts, algorithm design. Jiahong Chen contributed in part for the coding work of the illustrative simulation. Clarence W. de Silva was the supervisory author and involved throughout the research and manuscript composition. Vijay Bhargava provided some general guidance.

Chapter 5 is based on the publication [Tongxin Shu, Kevin Dsouza, Vijay Bharagava, Clarence W. de Silva, "Using Geometric Centroid of Voronoi Diagram for Coverage and Lifetime Optimization in Mobile Wireless Sensor Networks", *2019 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*. IEEE, 2019.]. Tongxin Shu was responsible for the research concepts, algorithm design and coding work. Kevin Dsouza contributed to proof reading of the manuscript. Clarence W. de Silva was the supervisory author and involved throughout the research and manuscript composition. Vijay Bhargava provided some general guidance.

Table of Contents

Abstractiii
Lay Summaryv
Prefacevi
Table of Contents viii
List of Tablesx
List of Figures xi
List of Symbols xiii
List of Abbreviations xiii
Acknowledgements xvii
Dedication xvii
Chapter 1: Introduction1
1.1 Motivation1
1.2 Problem Formulation
1.3 Research Objective
1.4 Related Work
1.4.1 Adaptive Sampling
1.4.2 Dual Prediction Scheme
1.4.3 Sensing Coverage Maximization in MWSN10
1.5 Contributions and Organization of the Dissertation
Chapter 2: Adaptive Sampling15
2.1 Data-Driven Adaptive Sampling Algorithm

2.2	Illustrative Simulation	
2	Simulation Results	
2.4	Model Validation	
Chaj	oter 3: Dual Prediction Scheme	
3.	Problem Formulation and Algorithm Development	
3.2	Dual Prediction Scheme in WSN	
3.	The LMS Algorithm	
3.4	The Long Short-Term Memory Recurrent Neural Network (LSTM RNN)	
3.:	Hybrid DPS Based on LMS and LSTM	
3.0	Simulation Study	
	3.6.1 Data Selection	
	3.6.2 Experimental Setup and Parameter Determination	
3.	Simulation Results	
Chaj	oter 4: Optimal Estimation of Sensor Nodes	66
4.	Sensing Models	
4.2	Number of Sensor Nodes	68
Chaj	oter 5: MWSN Lifetime and Maximum Sensing Coverage	72
5.	Assumptions	
5.2	Hole Detection Strategy Based on Voronoi Diagram	74
5	Hole Healing Strategy	
5.4	Illustraitive Experiment	80
Chaj	oter 6: Conclusion and Future work	85
Bibli	ography	88
		ix

List of Tables

Table 2.1. Results with different threshold values for DO data sampling	26
Table 2.2. Turbidity data sampling results for different threshold values.	29
Table 2.3. Turbidity data sampling results for different threshold values.	31
Table 2.4. Performance comparison between ASA and data-driven adaptive sampling algorit	:hm
(DDASA) using DO data	32
Table 2.5. Model validation using five-fold cross-validation	33
Table 3.1. The number of neurons in the 1 st layer and the corresponding error	54
Table 3.2. The number of neurons in the 2 nd layer and the corresponding error	55
Table 3.3. The prediction accuracy for different pre-defined error values	58
Table 3.4. Maximum absolute prediction error (in ° Celsius) for different pre-defined error	
values	60
Table 3.5. Energy consumption details for data transmission and sampling	63
Table 5.1. Realistic parameter with values.	82

List of Figures

Figure 2.1 Representation of the revised sigmoid function.	17
Figure 2.2. (a) Original dissolved oxygen (DO) Data; (b) Sampled DO data with $t = 0.01$; (c)	
Sampled DO data with $t = 0.015$; (d) Sampled DO data with $t = 0.02$	23
Figure 2.3. The differences among DO data for different sampling intervals	24
Figure 2.4. Frequency trend with $t = 0.009$, $t = 0.01$, $t = 0.015$ and $t = 0.02$	25
Figure 2.5. (a) Sampled DO data with $t = 0.03$ and $t = 0.07$; (b) Frequency trend with $t = 0.03$	
and <i>t</i> = 0.07.	26
Figure 2.6. (a) Original Turbidity data; (b) Sampled Turbidity data with $t = 0.110$; (c) Sample	d
Turbidity data with $t = 0.112$; (d) Sampled Turbidity data with $t = 0.115$	28
Figure 2.7. Reconstructed DO signal based on adaptive sampling algorithm (ASA)	31
Figure 2.8. Algorithm performance with respect to the remaining battery level	32
Figure 2.9. Wireless sensor network (WSN) setup in the Intel Berkeley Research Lab	35
Figure 2.10. (a) Original Temperature data from Intel Berkeley Research Lab; (b) Sampled	
Temperature data with $t = 0.0016$	36
Figure. 3.1. (a) A WSN with multiple clusters including CHs; (b) A WSN without CHs	40
Figure. 3.2. The process of implementing DPS between a sensor node and the CH	42
Figure. 3.3. Structure of the LMS algorithm	44
Figure. 3.4. Structure of the LSTM cell	47
Figure. 3.5. Hybrid DPS indicating when the model needs updating	51
Figure. 3.6. Model loss in 50 epochs of training	56
Figure. 3.7. Estimated data compared to original data when $e_{threshold} = 0.2$	58
Figure. 3.8. Number of transmissions for different pre-defined error values	61 xi

Figure. 3.9. Data packets sent in terms of different pre-defined error values	64
Figure. 3.10. Trade-off between accuracy and energy consumption for different threshold	
values	64
Figure. 4.1. (a) Binary sensing disk model; (b) Probabilistic sensing model	67
Figure. 4.2. First step of deploying sensor nodes to cover most space of a rectangular area	
	69
Figure. 4.3. Areas left uncovered after first round deployment	70
Figure. 5.1. The partitioned Voronoi polygons with nodes residing inside	75
Figure. 5.2. Node's movement for healing the detected hole	77
Figure. 5.3. Changes of the covered area when node <i>n</i> 1 moves	78
Figure. 5.4. (a) Randomly deployed nodes before repositioning; (b) Deployed nodes after	
repositioning	81
Figure. 5.5. Guaranteed node-to-node connectivity after repositioning	81
Figure. 5.6. Total distance travelled on average when different number of nodes are chosen	
	84

List of Symbols

convex polygon
corresponding coordinate of the sensor node
cell state of the LSTM cell
corresponding coordinate of the vertex
absolute difference between X_{i+1} and X_i over the average value of a
number of N sliding-window-based most recent data
desired signal to which the filter adapts
Euclidean distance between the node n_i and the vertex V_k
error between the desired signal and the prediction
pre-defined threshold for Hybrid DPS
constant sampling frequency
new sampling frequency
forget gate in the LSTM cell
input vector
input gate in the LSTM cell
number of equal sized subsets for conducting cross validation
number of sliding-window-based data
sensor node that resides inside the polygon
output gate in the LSTM cell
packets of beacon signals
packets of normal signals that contain the sensed data

P_T	total packets
S	the sequence of collected samples
t	pre-determined threshold for DDASA algorithm
V _k	<i>k</i> th vertex of the Voronoi polygon
W	weight vector of LMS filter
W _{opt}	optimal weight vector
W^{T}_{opt}	transpose of the optimal weight vector
x	predicted data using DPS
X(n)	input signal vector collected by a sensor node
X_i	<i>i</i> th sample in the sequence of collected data
у	predicted data using LMS filter
Y	revised sigmoid function
α	ratio of P_N and P_B
Е	Euclidean distance
η	learning rate

List of Abbreviations

ARIMA	Autoregressive Integrated Moving Average
ASA	Adaptive Sampling Algorithm
СН	Cluster Head
DDASA	Data-driven Adaptive Sampling Algorithm
DO	Dissolved Oxygen
DPS	Dual Prediction Scheme
FFT	Fast Fourier Transform
GC	Geometry Centroid
GCVD	Geometric Centroid Voronoi Diagram
GW	Gateway
LMS	Least Mean Square
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MSE	Mean Squared Error
MWSN	Mobile Wireless Sensor Network
NME	Normalized Mean Error
NOAA	National Oceanic and Atmospheric Administration
P(i)	<i>i</i> th Polygon within the ROI
RNN	Recurrent Neural Network
ROI	Region of Interest
VD	Voronoi Diagram

VP Voronoi Polygon

WSN Wireless Sensor Network

Acknowledgements

No words can ever describe my sincere gratitude to Dr. Clarence W. de Silva who always steered me into the right direction throughout my five years study at UBC. I wholeheartedly cherish and value the opportunity given to me to study and conduct research at the University of British Columbia, as well as the laboratories and facilities provided by Dr. de Silva. I would not come this far and achieve what I have done without his constant and enduring help and support over the past years. Also, I want to thank Dr. Vijay K. Bhargava and all my committee members for their valuable advice for making my completion of this thesis more thorough and complete.

This research project has been supported by the India-Canada Centre for Innovative Multidisciplinary Partnership to Accelerate Community Transformation and Sustainability (IC-IMPACTS) and by research grants from Natural Sciences and Engineering Research Council of Canada (NSERC), China Scholarship Council and the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation held by Dr. Clarence W. de Silva.

I am grateful to the current and past members of Industrial Automation Lab, and I especially wish to extend my gratitude to Dr. Teng Li, Dr. Jiahong Chen and Mr. Fan Yang, who have always inspired me and offered me help both in academic matters and in life. Also, I am thankful to all the other colleagues including Dr. Min Xia, Mr. Hani Balkhair, Mr. Zhuo Chen, Mr. Lucas Falch, Mr. Hiroshan Gunawardane, Mr. Kuangen Zhang, Mr. Jing Wang, Mr. Jayasanka Ranaweera, Mr. Duong Nguyen, Mr. KNR Surya Vara Prasad and Mr. Kevin Bradley Dsouza. Particularly, as always, my deepest gratitude goes to my parents, who have always been there, loving and supporting me unconditionally throughout my upbringing.

Dedication

I dedicate this thesis to my beloved parents and grandparents for their endless love and priceless sacrifices.

This work would not ever have been possible without their support.

Chapter 1: Introduction

1.1 Motivation

The water quality monitoring project, on which the present research is based, is primarily motivated by the fact that today people in rural and developing regions of the world are at a high risk of exposure to water-related diseases. However, problems of poor water quality are not limited to such regions. Urban areas in industrialized countries are equally vulnerable. For example, there were two major water-related crises that seriously affected Flint City, Michigan, USA in 2016 due to the leakage of lead in water distribution pipes, and Nova Scotia, Canada in 2019 because of the effluent leakage into a body of water. These problems could have been avoided and corrective actions could have been taken in a timely manner if a reliable, accurate, and distributed water monitoring system was available in the affected areas that could rapidly provide accurate information about the water contamination.

Developing a reliable and automated monitoring system consisting of dynamic sensor network for spatiotemporal monitoring is arguably challenging and of great significance. Accordingly, research on the subject of remote environmental monitoring has been prominent in recent years. One cannot overstate the importance of automated remote environmental monitoring, since it can result in convenience and flexibility of observing environmental conditions from distance, thereby reducing the risks, cost, and the required time while improving the accuracy and efficiency. Furthermore, it is then possible to monitor large areas in near real time and even predict some

natural disasters such as volcano eruptions, sandstorms, and earthquakes, and take remedial actions ahead of the catastrophe.

1.2 Problem Formulation

The limited resources of a WSN are the key factor that determines the lifetime and efficiency of the WSN. Despite the fact that there are some advanced energy-harvesting techniques, without proper energy management schemes, the electronic devices may still encounter energy shortages and thus become dysfunctional. Not limited to the situations where energy harvesting possibilities are slim, a scheme that improves the energy efficiency of a WSN will prolong the lifetime of the WSN and the limited energy supply can be properly utilized.

Intuitively, there always exists a trade-off between the energy consumption and the accuracy of the sampled data. For instance, a more frequent sampling activity will collect more data samples for the user but, as a consequence, more energy will be consumed. Similarly, a more frequent node-to-node data transmission will make more collected data available for use, but at the cost of increased energy consumption. In this regard, finding a balance between the data accuracy and the energy consumption remains an interesting and important research topic. In addition, for a given monitored area, the deployment strategy of a limited number of sensor nodes also plays a key role in deciding the working efficiency and energy consumption of the overall WSN.

With regard to the problem specified here, the strategies for addressing the power management issues can be categorized in terms of the network layers. For instance, the sampling frequency of

each single sensor node could be adjusted according to the sampling needs or environmental changes. This scheme is based on the level of a single node. Next, considering the overall sensor node overhead in a WSN, the investigation of optimal routing schemes for the sensor nodes is also a potential research topic that has drawn much attention [1]. Similarly, a reduced transmission overhead among sensor nodes also contributes to the energy conservation. That being said, a reduced number of transmissions among the sensor nodes leads to reduced energy consumption. Moreover, considering the spatiotemporal relationship among the sensor nodes, exploiting the geographical information in terms of the monitored environment will also lead to a problem of sensor deployment optimization [2]-[3]. With an optimal deployment of the same amount of energy for the sensing activities [4]-[5]. Likewise, a fewer number of sensor nodes would be needed if each of them could be deployed at an optimal location [4].

1.3 Research Objective

With the technical and potential research issues mentioned above, this thesis investigates and develops energy efficient schemes for effectively prolonging the lifetime of a WSN and simultaneously ensuring high data accuracy. The schemes are developed on multiple layers of the sensor network, which means they should be flexible for implementation in other practical environmental monitoring applications. The main research objectives of the present thesis are as follows: 1. On the sensor node level, develop an energy efficient scheme that neutralizes the energy consumption and the accuracy of the collected data, which entails that the energy consumed within

each single sensor node should be conserved as much as possible while the data accuracy of the sampled data is guaranteed. 2. On the node-to-node communication level, develop a dual prediction scheme (DPS) that reduces the necessary number of transmissions. 3. With a limited number of sensor nodes in a Mobile Wireless Sensor Network (MWSN), design a scheme that minimizes the total moving distance of all the sensor nodes and also maximizes the total sensing coverage within the monitored field.

1.4 Related Work

The subject of power management has been investigated by others in view of its importance in various applications. The power management in WSN is a broad topic and can be studied based on various aspects. The general methodologies for managing power in a WSN can be generally categorized as hardware design, software design, network protocols, and middleware services. The major energy-saving schemes designed under these four categories are mainly: radio optimization; battery repletion; sleep/wakeup schemes; energy-efficient routing, and; data reduction [6]. In terms of radio optimization, the traditional approaches focus on controlling the power used for signal transmission [7], where the nodes in a WSN require knowledge of the power levels and link qualities of their neighbor nodes. Hence, the designed schemes have a spatiotemporal impact on the wireless sensor network, either locally or globally. With regard to battery repletion, in recent years applications have been broadly developed for remote environmental monitoring where the monitored environment is not easily accessible. Considering that it might be time- and manpower-consuming to replace the batteries in a remote sensor network, some work that focuses on the techniques of energy harvesting and wireless charging has been carried out [7]. Typical among the

sleep/wake schemes is duty cycling. This scheme is usually categorized as on-demand, asynchronous, and scheduled rendezvous. In [8], a hybrid method based on the battery state and the stability of water quality has been proposed. The nodes can be switched on or off either depending on its remaining battery state or the need of sampled data. Moreover, in energy-efficient routing techniques, the distance between each regular node and also the sink node of the WSN plays a key role. Hence, in order to optimally find a routing path that is energy efficient, schemes relying on single path and multiple paths have been proposed. A survey paper on this subject is available [9].

1.4.1 Adaptive Sampling

Data reduction exploits the fact that depending on the characteristics of the sampled data within the environment, some data could be redundant. For example, in water quality monitoring, if the monitored parameters remain relatively stable within the time period of interest, it is believed that no significant changes will happen. Thus, a relatively low sampling frequency may be used for the sake of saving energy. With fewer sampled data, the trend of the water quality can still be analyzed. This is the main advantage of using adaptive sampling for power management. Besides data reduction, there are some other similar approaches such as data aggregation and data compression [2] that also contribute to energy conservation by using fewer but representative data.

It may be desirable to predefine a standard for the quality of data when applying adaptive sampling techniques. Drira et al. [10] developed a location-aware scheme for an adaptive data collection system in vehicular networks, which constantly compares the travel time of a vehicle with a predefined value for deciding whether to transmit messages between a traffic management centre

and a vehicle. It is found that their developed algorithm is capable of reducing the communication load and data storage requirement while maintaining a high accuracy for estimating the fuel consumption and emission. Analogously, to ensure the quality of data, the proposed data-driven adaptive sampling algorithm (DDASA) constantly compares the latest sampled data with a set of historical data to determine a new sampling frequency. However, the energy consumption in their methods is reduced primarily by limiting the number of transmissions, while the approach in the present thesis focuses on saving energy on the sensor node level.

Prabha et al. [11] proposed a context aware sensing technique, which could be utilized for landslide monitoring. According to their approach, given a set of data, a discrete wavelet transformation is performed to determine the lowest sampling rate, which meanwhile should guarantee the reliability of data. Based on the characteristics of data, three level thresholds are set to derive the sensor/network level contexts as Safe, Listen and Alert. The system initially starts to function at the lowest sampling rate until the sensor/network level contexts change (i.e., from Safe to Listen, or Listen to Alert). Hence, the sampling interval would be dynamically varied depending on the sensor/network-level contexts. As a result, the energy for sensory tasks could be saved. While their strategy is conceptually similar to the one developed in the present thesis, they only consider three fixed sampling intervals (i.e., sampling frequencies), which need to be predefined manually. It is believed that much more energy could be saved if the scheme for changing sampling frequency could be data-driven, rather than based on three predefined sampling intervals.

In addition, an adaptive sampling approach for snow monitoring applications has been developed in [12]. This method initializes the sampling rate by conducting a fast Fourier transform (FFT) on a sequence of pre-sensed data to acquire the maximum frequency, and depending on the subsequent data, a new maximum frequency is determined. By comparing the variation of the current sampling rate and the new maximum frequency, a new sampling rate is established. However, this algorithm highly relies on a large number of pre-sensed data, and also changes the sampling rate only depending on the FFT of the newly sampled data to avoid under-sampling. Nonetheless, in practical environmental monitoring situations, under-sampling is acceptable if the frequency of the sensed data hardly fluctuates and remains relatively steady within a period of interest. In this manner, in order to dynamically change the sampling frequency, properly using the actual value differences and stability of the sensed data makes more sense than simply using an FFT of the subsequent data. This is essentially what makes DDASA different from a traditional ASA.

In [13], an adaptive sampling scheme for acquiring wind data is proposed based on energy awareness. Rather than being data accuracy oriented, this algorithm is essentially energy awareness oriented. This means when the remaining energy of the battery drops below a predetermined threshold, the sampling frequency will be decreased accordingly. The larger the gap between the remaining battery state and the predetermined threshold, the faster the drop of the sampling frequency, in order to maximize the life of the battery. Apparently, much more energy could be saved when the battery state is relatively low, through this proposed scheme. However, since there is always a tradeoff between the accuracy of the sensed data and the energy conservation, a strategy that is primarily driven by the battery state will surely compromise the accuracy of the sensed data, and possibly under-sampling can occur as a result. Hence, it is believed that a scheme that is data accuracy-driven, would be more reasonable and universally applicable.

1.4.2 Dual Prediction Scheme

In the context of energy conservation in WSN, the prevailing approaches of data reduction have to be explored. The DPS, which serves as an energy-efficient scheme in the network-level communication, features data reduction and transmission reduction by conducting time-series prediction. While the work in [14] proposed the DPS and examined its performance on a temperature dataset, a more recent work [15] statistically proved the significance and efficiency of implementing DPS for transmission reduction.

Since the DPS is closely associated with time-series prediction methods, especially when the collected data exhibits a temporal correlation, a solid prediction method should be developed in order to provide reliable prediction results. When choosing a proper method, the internal characteristics of the WSN should also be considered, such as the overload and memory capacity of the entire network. An early and classical solution provided by [16] handled this issue by continuously updating the coefficients of the kernel linear regression model instead of transmitting real measurements.

The LMS is a light predictor and requires low memory and low computational complexity. The work in [14] proposed to use LMS as a predictor, which lays the foundation for conducting DPS. The results proved that by choosing the proper step size, the algorithm could effectively reduce

the transmission times and thus reduce the associated energy consumption. The coefficients of the algorithm were intimately based on a fine-tuning process and a specific data set. Thus, it is important that an improved LMS algorithm be developed. Especially when dealing with any given data set in an environmental monitoring application, the algorithm should be capable of adapting itself to the new data set and automatically determining an optimal step size during the prediction process. In this manner, the work in [17] provided a useful approach for dynamically changing the step size. The way of updating the step size in that work is to divide the old step size value by a fixed number M, which is determined by the dimension of the inputs. It was observed that the prediction accuracy would be acceptable since a large step size will boost the convergence speed in the early training phase and a smaller step size will ensure that the algorithm eventually converges in a globally optimal manner. However, it is speculated that when a higher dimension of the inputs is needed, the range of M becomes larger accordingly. Then it may take many rounds of trial to determine a suitable value for *M*. Besides LMS, there are other work such as [18][19][20] related to time-series prediction methods. For example, the use of Hidden Markov Model and Kalman filter can provide substantial prediction results. Nonetheless, the training progress is extensive and computationally expensive.

It is generally inferred that a reasonable combination of two methods of prediction, forming a hybrid method, could enhance the prediction accuracy. However, the predictors themselves should, to some extent, mutually benefit or collaboratively contribute to obtain a better prediction result, when two independent methods are hybridized. The work in [21][22] offer such a possible solution by combining the Autoregressive Integrated Moving Average (ARIMA) model and a Neural Network (NN) for improving the prediction accuracy. Similar to LMS, the ARIMA could capture

a linear relationship among adjacent data. However a feed-forward Neural Network, despite its capability of capturing nonlinear relationships among data sets, still cannot capture a temporal correlation among data. This would cause the prediction results either under-fit or over-fit. In this context, the LSTM is a promising predictor, which is able to address the relevant issues. Some recent work [23][24][25] has proved that the LSTM could provide satisfactory results for capturing the temporal relationship among data. Yet it has not been widely used in WSN applications.

It is known that for the present problem, either the existing methods are either computationally expensive or there exists much room to improve the prediction accuracy in choosing a better prediction model. The work in the present thesis provides a hybrid DPS that combines an improved LMS algorithm and the LSTM RNN. In seeking a trade-off between the algorithm complexity and efficiency, the present research also offers flexible solutions for leveraging the limited energy resources in a WSN.

1.4.3 Sensing Coverage Maximization in MWSN

The schemes for sensor node repositioning can be categorized primarily as on-demand-based and post-deployment-based [26]. The former one decides how some sensor nodes are relocated during the deployment phase, following certain demands such as the application-level needs or the network-level energy, while the latter one focuses on adjusting the locations of the sensor nodes after the deployment phase. As for the factors that have to be taken into account while relocating the sensor nodes, the coverage and the uniformity of the entire network need to be considered.

Also, since the energy used for propelling the mobile sensor nodes is significant, the time and distance that a node needs to travel also have to be taken into consideration.

The notion of virtual force has been proposed and utilized for node repositioning [27][28][6], which resembles the repulsive or attractive force among molecules, trying to establish a balanced distance for sensor repositioning. It is proved that the nodes can be evenly distributed in the ROI with a good uniformity; however, since there would always be sensors moving back and forth, in reaching an equilibrium, energy would be unnecessarily wasted due to the perturbation. Apart from this, based on the VD, three different types of movement-assisted schemes: VEC (Vector-based), VOR(Voronoi diagram-based) and Minimax are developed for coverage maximization [29]. Nonetheless, energy would still be wasted due to the redundant movements of the sensors. Particularly, a few nodes might lose connection with other nodes during the relocation process, which is undesirable in practical environmental monitoring applications. Based on VD, Xia et al. [30] propose to move the sink instead of the nodes in the overall MWSN, thus allowing the sink to selectively visit candidate vertices for data collection. In [31], a game theory-based algorithm has been designed to relocate the mobile robots in an unknown environment. The marginal sensing contribution, combined with the energy consumption due to the movement, are considered as part of an objective function. It is demonstrated that the designed algorithm yields a satisfactory result with a few number of mobile agents in a limited space.

1.5 Contributions and Organization of the Dissertation

The main contributions of this dissertation are listed below.

1. A data-driven adaptive sampling algorithm (DDASA) for node-level sampling is presented. It is proven that this algorithm is robust for different types of parameter sampling, and can effectively conserve energy with a satisfactorily reconstructed signal. Compared with some existing adaptive sampling algorithms, which are battery-state driven, there are justifiable occasions where the sampling frequency is based on the real-time sampled data, especially when the environmental parameters fluctuate, is of significant interest. Additionally, it is shown that, by dynamically changing the sampling frequency according to the newly sampled data, the proposed DDASA will outperform a traditional ASA with respect to the accuracy of the reconstructed signal and the energy conservation. Thus, the goal of prolonging the life of the nodes is achieved with the developed approach.

2. A Hybrid dual prediction scheme (DPS) is developed, which aims to make time-series prediction in environmental applications where a wireless sensor network is utilized. In the two-way communication between a normal sensor node and a cluster head (CH), the developed algorithm effectively reduces the number of necessary transmissions by forecasting the future data bilaterally. Meanwhile, the prediction accuracy can be maintained according to the error threshold that is predefined by the user. 3. Proposes a novel approach based on Voronoi diagram and its geometric center to determine the optimal location for sensing coverage maximization while ensure the connectivity among sensor nodes with least movement of the nodes. The proposed algorithm, GCVD, outperforms the existing algorithms that address the same problem, with respect to the energy conservation.

The rest of the dissertation is organized as follows.

Chapter 2 presents a novel adaptive sampling algorithm, which is a data driven scheme. Focusing on the environmental changes that numerically reflect on the sampled data, the sampling frequency is changed accordingly. With a dynamically changing sampling frequency, the energy consumed for the sampling tasks will be conserved. Meanwhile, the developed algorithm outperforms the existing methods and a fixed frequency sampling strategy, in terms of the energy conservation and data accuracy.

Chapter 3 develops a hybrid dual prediction scheme, which achieves reduced communication by the means of bilateral prediction between nodes. A normalized least mean square filter (LMS) is first used for data prediction since it has low computational complexity and requires reduced memory. If the predicted data fails to meet the accuracy requirement, then the long short term memory approach intervenes to make a data prediction since it can capture the temporal relationship as well as the nonlinearity characteristics among the collected data. Chapter 4 demonstrates a solution to another critical issue, which is how to efficiently deploy a limited number of sensor nodes in the monitoring field in order to maximize the sensing coverage. Considering the fact that the sensor movement consumes a significant amount of energy compared to what is consumed for sensing tasks and node-to-node communication, minimizing the total movement distance while maximizing the total sensing coverage is a key goal. In addition, in order to maintain the connectivity while the nodes are moving, the proposed scheme has to incorporate multiple factors for optimization.

Chapter 5 concludes the work and discusses possible future work.

Chapter 2: Adaptive Sampling

Ideally, the data sampling rate that is used for a sensor signal should depend on the rate at which the signal changes. The energy consumption for signal acquisition, processing, and transmission all depend on the sampling frequency, either directly or indirectly. Hence, for energy conservation, it is desirable to reduce the quantity of sampled data when the conditions (e.g., the water quality, in the present application) remain relatively stable. Meanwhile, if certain parameters in the process (water in the present application) are changing abruptly, the sampling frequency should be increased in order to acquire sufficient information about the condition of the process. It should be noted that not only will the sampling frequency have a significant impact on the energy usage, the processing and transmitting of the sampled data will also consume extra energy. With this in mind, a data-driven adaptive sampling algorithm (DDASA) is developed in the present chapter, which dynamically changes the sampling frequency based on the nature of the sampled signal. This algorithm would be universally applicable with respect to conserving energy and prolonging the lifetime of a WSN. This approach is not limited to water quality monitoring and is applicable in other types of monitoring applications using WSN, such as indoor environmental monitoring [32], structural health monitoring [33], climate conditions monitoring [34], and healthcare monitoring [35].

2.1 Data-Driven Adaptive Sampling Algorithm

The core idea behind the proposed DDASA is the design of a data accuracy-driven strategy of power utilization, which is particularly applicable in water quality monitoring, using autonomous

sensor nodes, depending on the environmental changes. It is clear that a higher sampling frequency is desired where the water quality is vulnerable to rapid environmental change or pollution, particularly in a post-industrial city. Particularly, the researchers and water quality observers of such areas might be more concerned and sensitive to the sudden changes of certain key parameters in the water. Correspondingly, if the monitored parameters hardly fluctuate, meaning no significant changes are taking place, it is believed that a lower sampling frequency is preferred, and as a result, less energy will be needed for data sampling, data processing and transmission.

Based on these assumptions, a revised sigmoid function is proposed for incorporation in the present DDASA, which is expressed as:

$$Y(D) = \frac{2}{1 + e^{-(D-t)}}$$
(2.1)

$$D = \frac{|X_{i+1} - X_i|}{\frac{1}{N} \sum_{i-N+1}^{i} X_i}$$
(2.2)

Here, *t* is a pre-determined threshold, and *D* is the absolute difference between X_{i+1} and X_i over the average value of *N* sliding-window-based most recent data. This function is used to dynamically change the sampling frequency. Since we are interested in knowing whether there exists a sudden environmental change or not, it is reasonable to compare the latest sensed data X_{i+1} with the former data X_i in the signal sequence, and then divide the absolute difference between them by the mean value of the most recent *N* data. If *D* is sufficiently large, it indicates a sudden environmental change. Then a somewhat higher sampling frequency is desired. Additionally, if the value of *D* is smaller than the threshold *t*, which means the value changes are not adequately significant, the

sampling frequency can be reduced in view of the relative stability of the monitored data. Hence, the theoretical value of *Y* is actually smaller than 2 but greater than *Y*(0), that is, since the smallest value of *D* will not be a number smaller than 0 in the data sensing process, the value of *Y*(0) is essentially greater than 0 regardless of the value of *t*. A representation of the revised sigmoid function is presented in Figure 2.1. It is found the value of *Y* in simulation studies is mostly a number either slightly smaller than 1 (e.g., when *D* equals to D_1) when the sensed data numerically remain stable or greater than 1 (e.g., when *D* equals to D_2) when the sensed data abruptly change.



Figure 2.1 Representation of the revised sigmoid function.

Since the value of Y(D) dynamically changes depending on the latest sampled data, the sampling frequency should also be changed accordingly. If the current sampling frequency is denoted by

 f_{curr} , which is used to acquire the latest data, then the new sampling frequency, denoted by f_{new} is given by:

$$f_{new} = f_{curr} \times y(D) \tag{2.3}$$

Hence, a new sampling frequency for the next iteration is a function of the newly sampled data and the average value of the latest N sliding window-based data, which satisfy the desirable requirements of a reasonable sampling frequency and energy conservation.

A pseudo code for implementing the DDASA scheme is given in Algorithm 1.
Algorithm 1. DDASA

- 1. Initialize a constant sampling frequency denoted by f_{const} , sample N number of samples for later use; store the samples in a sequence as S;
- 2. Predetermine a threshold *t* according to the characteristics of the monitored parameter;

3. Define
$$D = \frac{|X_{i+1} - X_i|}{\frac{1}{N}\sum_{i=N+1}^{i} X_i}$$

- 4. Define $f_{curr} = f_{const}$;
- 5. for (i=N; i++) {
- 6. Sample X_{i+1} based on f_{curr} (or f_{curr}');

7.
$$D = \frac{|X_{i+1} - X_i|}{\frac{1}{N} \sum_{i=N+1}^{i} X_i};$$

8.
$$Y(D) = \frac{2}{1 + e^{-(D-t)}}$$
;

9.
$$f_{new} = f_{curr} \times Y(D);$$

- 10. $f_{curr}' = f_{new};$
- 11. $S(i+1) = X_{i+1};$
- 12. end
- 13. return *S*;

It should be noted that the use of a sigmoid function corresponds with both the physical nature of the sampling process and the sensed data. The present scheme ensures that a new sampling frequency is adjusted not just based on the latest sensed data, but also on a set of data from a previous period, which reflect the overall environmental conditions throughout the operating time. More precisely, a faulty reading, whose value might be unusually higher or lower than the average, would not change the next sampling frequency drastically. Instead, if an abrupt change is detected, which means a certain reading numerically increases or decreases quickly, the future sampling frequency will be increased gradually based on a consecutive set of obtained data. Equivalently, a much higher sampling frequency is the consequence of using f_{curr} to multiply Y(D) (greater than 1 but lower than 2) in each iteration for multiple times after comparing the latest acquired data with a set of past-period data. As a result, this data-driven scheme allows the energy to be reasonably either consumed or conserved, as it is the trend of the sensed data rather than certain unusually high or low value data that decides the future sampling frequency.

2.2 Illustrative Simulation

The performance of the proposed algorithm is assessed by using a set of real-time data provided by the National Oceanic and Atmospheric Administration (NOAA) [25]. NOAA uses distributive platforms (buoys) to form a network for collecting real-time water-quality data. The data chosen for the simulation are collected from a platform called "Jamestown," where the monitoring duration was from 15 December 2016 to 15 March 2017 with a sampling interval of 1 h for a sample. To validate the robustness of the DDASA, two distinct parameters, turbidity and DO, are selected since their value range differs enormously among various water-related parameters. In order to evaluate the performance of the scheme for different values of pre-determined parameters, the term Normalized Mean Error (NME) is introduced, which indicates the overall goodness of fit, and is defined as:

$$NME = \frac{1}{n} \sum_{i=1}^{n} |\hat{x}_i - x_i| \times 100\%$$
(2.4)

2.3 Simulation Results

The simulation results are divided into two parts. First, the proposed DDASA is tested using DO and turbidity data separately, followed by the selection of a list of parameters and the corresponding performance indicator (i.e., NME). Later, with regard to energy conservation, a comparison between the algorithm proposed in the present research and a traditional adaptive sampling algorithm is presented. The simulation results demonstrate that the DDASA algorithm is not only capable of maintaining a high level for energy-efficient sampling, but also has the ability for effectively reconstructing the original signal with much fewer data.

A plot of the original DO data signal is shown in Figure 2.2(a) for later comparison, which consists of 2182 sensed samples in total. In interpreting the original signal, it may be inferred that an abrupt change in the water quality occurs around the 600th sample, leading to a significant increase of the DO content.

In the simulation, the initial sampling frequency is set to the same value as the constant frequency used for sampling the 2182 data. The window size (i.e., N) is set to 50, while the threshold is a variable. Then a linear interpolation between two neighboring measurements is used, to fit the reconstructed signal with respect to different threshold values. Hence, the trend of the DO parameter is given intuitively.

In Figure 2.2(b)–(d), the simulation results with different threshold values are presented. It is seen that as the predetermined threshold increases, fewer sample data are acquired based on the DDASA. Combined with the linear interpolation fitting, the trend of the DO content can still be maintained at a high level of similarity compared to the original DO signal when t = 0.01, t = 0.015 and t = 0.02.



Figure 2.2 (a) Original dissolved oxygen (DO) Data; (b) Sampled DO data with t = 0.01; (c) Sampled DO data with t = 0.015; (d) Sampled DO data with t = 0.02.

To provide a more intuitive understanding when adopting different sampling intervals, the difference among the sampled DO data is shown in Figure 2.3 correspondingly, which is the outcome of a combination of Figure 2.2(a)–(d).



Figure 2.3 The difference among DO data for different sampling intervals.

The frequency trend for each corresponding threshold is also shown in Figure 2.4. It is believed that when the threshold is set to a proper number, (e.g., t = 0.01), the sampling frequency will eventually converge or remain at a stable level. However, it is also found that when the threshold is set to a value smaller than 0.01, for instance, when t = 0.009, the frequency trend will not converge. This happens mainly because the value of D(i) in most iterations is greater than the threshold, which results in a constant increase in the sampling frequency. This also means the dynamically changed sampling frequency is not sensitive to the data change, which in fact is not a desired outcome. Additionally, when t = 0.015 and t = 0.02, compared to the case when t = 0.01,

the sampling frequency is more sensitive to the data change, and as a consequence, more energy would be conserved since the sampling frequency in each iteration is lower than the initial sampling frequency.



Figure 2.4 Frequency trend with t = 0.009, t = 0.01, t = 0.015 and t = 0.02.

In Table 2.1, an investigation of the performance of the algorithm and the predetermined threshold values is given. It is noticed that although the value of the predetermined threshold increases evenly, the number of samples drops unevenly; however, the NME increases as the number of samples decreases. In Figure 2.5, it is clear that some important peaks are missing due to a lack of sampled

data, where the threshold equals 0.03 and 0.07. The corresponding NME are 9.99% and 11.70%, which is relatively high. Thus, in order to find a suitable value for the threshold, it would be necessary to preset a value for the minimum required data quality in terms of NME and, meanwhile, the threshold value should not be too small, which ensures that the sampling frequency will eventually converge. More details regarding the choice of the threshold value will be given later.

 Table 2.1. Results with different threshold values for DO data sampling.

	<i>t</i> = 0.01	t = 0.015	t = 0.02	t = 0.03	t = 0.07
Number of Samples	1064	548	421	297	146
Normalized Mean Error (NME)	1.62%	5.52%	8.43%	9.99%	11.70%



Figure 2.5 (a) Sampled DO data with t = 0.03 and t = 0.07; (b) Frequency trend with t = 0.03 and t = 0.07.

A similar simulation is conducted based on the turbidity data. The results are presented in Figure 2.6 with threshold values of 0.110, 0.112 and 0.115. A comprehensive comparison is given in Table 2.2. The same linear interpolation scheme is implemented between two neighboring measurements to fit the plot.



Figure 2.6 (a) Original Turbidity data; (b) Sampled Turbidity data with t = 0.110; (c) Sampled Turbidity data with t = 0.112; (d) Sampled Turbidity data with t = 0.115.

	<i>t</i> = 0.110	<i>t</i> = 0.112	<i>t</i> = 0.115	t = 0.120	<i>t</i> = 0.140
Number of Samples	1591	422	320	244	172
NME	1.14%	4.26%	5.33%	5.39%	6.16%

Table 2.2. Turbidity data sampling results for different threshold values.

It can be shown that when the threshold value increases from 0.110 to 0.112, the number of samples drastically drops from 1591 to 422, while an NME of 4.26% is still tolerable when t equals 0.112. As t increases from 0.112, the number of samples decreases correspondingly, which is, however, not as many as in the case when t increases from 0.110. Thus, it is reasonable to believe that setting the threshold value as 0.112 is a suitable choice, while choosing a value of 0.115, 0.120 or 0.140 for t is also acceptable considering their corresponding NME. Since only 422 or fewer data values are sampled with the DDASA, which is less than quarter of the original data values, the energy consumed for data acquisition, storage and transmission would be significantly reduced. Thus, the lifetime of the battery will be prolonged while maintaining a high level of accuracy within the sampled data, compared to the original ones.

A comparison of the algorithm performance is presented in Figures 2.7 and 2.8, and Table 2.4. Using the same sequence of DO data, a traditional ASA is implemented as well. A set of parameter values are chosen as empirically suggested in paper [12]. Figure 2.7 shows that the reconstructed signal using ASA can still indicate the fluctuation of the DO compared to the original signal in Figure 2.2(a). The NME for the results is 5.31% with 637 samples. The performance is similar when t = 0.115 for analyzing DO data in the proposed DDASA. However, to achieve a similar result, the ASA uses 637 samples in total, which proves to be more energy-consuming in contrast to the proposed DDASA, where only 320 samples are needed. Figure 2.8 presents the comparison of energy consumption over time for ASA and for different values of predetermined thresholds of DDASA. Considering the fact that a more frequent sampling process will correspondingly increase the energy consumption for storing and transmitting the data, to simplify the comparison process, only the energy consumed for keeping a node active and for data acquisition is considered. Moreover, the active time of the node in each case will be equivalent, which means it should be equal to the total length of time for a fixed rate sampling process. The specific quantity of energy consumption for different sensing activities can be found in [36]. Table 2.3 lists all the detailed information concerning the energy usage, which is the average cost for the Waspmote sensor node that was used for conducting the present practical experiments [36]. The battery has a capacity of 66000 mAh with 3.7 V nominal voltage. To simplify the process of performance comparison, we compare the proposed DDASA with the ASA in terms of the remaining energy, in which only the energy for

a node to remain active and collect data samples is considered. That is:

$$W_{remain} = W_{total} - W_{sample} - W_{active}$$
(2.5)

where W_{remain} , W_{total} , W_{sample} and W_{active} stand for the remaining energy, total energy, sampling energy and the energy for the node to remain active, respectively, which are expressed in joules.

The sampling interval of the original DO data is 1 hour. The simulation results are presented in Figure 2.8. The battery level given in *y* axis is calculated as W_{remain}/W_{total} . In Table 2.4, a performance comparison is provided. It is seen that the DDASA outperforms the ASA when t = 0.115 and t = 0.112, with respect to the remaining battery level.

Activity	Power (mW)	Time (ms)	Energy (µJ)
Transmitting data (16 bytes)	108.9	0.5	54.45
Receiving data (16 bytes)	59.4	0.5	29.70
Sampling DO	38	1600	60,800
Staying active	0.858	N/A *	N/A
Sleeping	0.0129	N/A *	N/A

 Table 2.3. Average power and energy consumption for different activities.

* The duration of time can be adjusted based on the sampling needs.



Figure 2.7 Reconstructed DO signal based on adaptive sampling algorithm (ASA).

To focus on the performance of adaptive sampling algorithms and simplify the process of algorithm comparison, it is assumed that there is no data transmission that costs extra energy. Also,

the sensor node remains active continuously without any sleeping activity. Then, a performance comparison is conducted in terms of the remaining battery level in Figure 2.8.



Figure 2.8 Algorithm performance with respect to the remaining battery level.

Table 2.4. Performance comparison between ASA and data-driven adaptive sampling algorithm (DDASA) using DO data.

	DDASA (<i>t</i> = 0.115)	DDASA (<i>t</i> = 0.112)	ASA	DDASA (<i>t</i> = 0.110)	Fixed Rate Sampling (f = 1/3600 Hz)
Number of Samples	320	422	637	1591	2182
NME	5.33%	4.26%	5.31%	1.14%	0
Remaining Battery Level	86.03%	82.86%	80.72%	75.19%	55.37%

2.4 Model Validation

In the scheme developed in the present work, finding an appropriate value for the threshold plays a key role when determining the performance of the algorithm. The value of D in Equation (2.3) is highly related to not only the absolute value between the newest data sample and the former one, but also the mean value of a set of window-based data. Thus, the function for determining a suitable threshold value should follow suit, which is defined as:

$$t = \frac{\frac{1}{k-1}\sum_{i=2}^{k} |x_i - x_{i-1}|}{\frac{1}{k}\sum_{i=1}^{k} x_i} = \frac{k}{k-1} \cdot \frac{\sum_{i=2}^{k} |x_i - x_{i-1}|}{\sum_{i=1}^{k} x_i}$$
(2.6)

To validate the proposed model, a *k*-fold cross-validation is conducted based on Equation (2.6). To simplify the cross-validation process, a five-fold cross-validation is selected for the total number of data. Meanwhile, the entire set of data is divided into five equal sized subsets, four of which are utilized as the training sets while the remaining subset functions as the testing set. In each cross-validation process, a threshold value is given depending on Equation (2.6), in which X_i denotes the sensed data amongst the training set. Afterwards, based on the threshold value, an NME is determined by comparing the reconstructed signal against the training set. If each of the five subsets is denoted by A, B, C, D and E, separately, the outcome of the cross-validation is presented in Table 2.5.

Training Sets	ABCD	ABCE	ABED	ACED	BCED
Testing set	Е	D	С	В	А
Threshold	0.0102	0.0113	0.0115	0.0120	0.0136
NME	3.37%	3.40%	3.12%	2.53%	2.25%

 Table 2.5. Model validation using five-fold cross-validation.

It is found that through the cross-validation process, the average value of NME is 2.93% while a suggested average value of the threshold is 0.01172. This threshold value and its corresponding NME also correspond with the results presented in Table 2.1, where a slightly smaller value of t results in a smaller NME, while a slightly larger value of t leads to a higher NME.

To demonstrate that the algorithm developed in the present work is widely applicable, a different dataset from Intel Berkeley Research lab [37] is utilized. The dataset is comprised of real-time data for temperature, humidity, light and voltage measured by Mica2Dot sensor nodes within a WSN (see Figure 2.9). In particular, a set of temperature data consisting of 25,000 samples, which were collected by Sensor 13 during 28 February 2004 to 20 March 2004, is selected as the original data set. Based on the entire data set, a suggested threshold for sampling is derived using Equation (2.6), which is 0.0016. The reconstructed signal is shown in Figure 2.10, which has an NME equal to 0.08%, and 17,957 samples were needed. Since around 7000 samples are deducted by comparing with the original data set, it is believed that considerable energy can be saved throughout the sampling process. Furthermore, since the accuracy of the reconstructed signal is 99.2% when *t* equals 0.0016, for further saving energy, slightly increasing the value of *t* and thus reducing the sample numbers should also be considered. For instance, when *t* increases to 0.0020, the corresponding NME is 0.29%, while only 11,423 samples are necessary.



Figure 2.9 Wireless sensor network (WSN) setup in the Intel Berkeley Research Lab.



Figure 2.10 (a) Original Temperature data from Intel Berkeley Research Lab; (b) Sampled Temperature data with t = 0.0016.

The DDASA algorithm of the present work focuses on addressing the energy consumption issues on the sensor node level, which is efficient in conserving energy for "energy-hungry" sensor nodes. However, saving energy from the aspect of reduced communication would be another avenue for proper power management. Such an energy-efficient methodology is presented in Chapter 3.

Chapter 3: Dual Prediction Scheme

Saving energy by reducing the node-to-node communication could be equivalently crucial in conserving energy, as adopting an adaptive sampling algorithm. They are both done at the sensor node level. This chapter develops a relevant communication scheme.

Typically, environmental monitoring is done over a geographic area. Hence, the measurements will demonstrate a spatiotemporal correlation, despite the fact that some of them may change in an imperceptible manner. In addition, some observations on temperature, humidity, wind, and rainfall, and so on are seasonally affected, which means there might be an ascending or descending trend starting from a certain time point in a year. In this context the correlation of the measured data can be rather obvious. As a result, the data reduction of those environmental measurements can be realized and consequently, only a lower extent of communication would be necessary [38].

An efficient way to carry out data reduction in a WSN is through data prediction. Based on the correlation among the measurements, data prediction can facilitate the sensor nodes to provide the latest environmental information without necessarily conducting the sensing task. As a result, the total degree of communication will correspondingly decrease, which paves the way for further energy conservation within a WSN.

In the grand scheme of things, the Least Mean Squares (LMS) filter is a useful tool, which lays the foundation for implementing a Dual Prediction Scheme (DPS). The LMS filter itself has been broadly used in various applications [39][40]. Due to its low computational complexity and 37

reduced storage requirements, the LMS filter is expected to be beneficial in WSN. Meanwhile, the DPS is data-quality based, allowing both the sensor nodes and the gateway (GW) or cluster head (CH) to predict the data contemporaneously. Only when the data quality is undesirable, meaning that the error between the predicted data and the real sensed data exceeds a pre-defined threshold, the sensor nodes will send the sensed data to the GW/CH and consequently will update the coefficients of the filter. While the sensor nodes are selectively sending the measurements to the GW/CH, further energy savings are realized as there is reduced level of data transmission as a result.

Moreover, the sensed signal (i.e., the time series data) in environmental monitoring might have linear and nonlinear segments. Although the LMS has an outstanding performance in predicting time-series data, the filter itself treats each segment as a piecewise linear function; hence, an autoregressive process lays the foundation for the future value prediction. Thus, it is believed that a tool for predicting the nonlinear segment will strengthen the outcome of the data prediction. In this regard, a Long Short-Term Memory Recurrent Neural Network (LSTM RNN) would be a suitable option.

In the present work, a normalized LMS algorithm is utilized. It is such a scheme that is designed under the DPS, which allows the desired signal to replace the unsatisfying prediction results. Also, considering the nonlinearity of the sensed signal, an LSTM RNN is combined with the normalized LMS in order to further compensate for and improve the performance of data prediction.

3.1 Problem Formulation and Algorithm Development

In most environmental monitoring applications, a WSN typically consists of dozens of sensor nodes, which are spatially deployed over a geographic area of interest. Based on the geographical condition and the monitoring goals, the specific number of nodes needed may vary accordingly. However, the structure of the typical WSN is generally categorized into two types: 1) a WSN with multiple clusters; 2) a WSN without any clusters. As shown in Figure 3.1(a), in a clustered sensor network, a cluster head (CH) plays the role of communicating with its surrounding sensor nodes and processing the received data by techniques such as data fusion and data integration. Afterwards, the processed data will be directly sent from the CH to the GW, ready to be analysed by the WSN host. Alternative simpler case would be a WSN without any CH taking the responsibility of gathering data and pushing them forward to the GW. Instead, following a routing protocol (see for example, [41], [42], [43]), some sensor nodes within the WSN would directly transmit the sensed data to the GW, which makes the entire network assume a flat organizational structure. By means of GPRS/3G/Radio communication, the acquired and pre-processed data can be uploaded and later analysed at the user end.



(a)



(b)

Figure 3.1 (a) A WSN with multiple clusters including CHs; (b) A WSN without CHs.

3.2 Dual Prediction Scheme in WSN

Regardless of the structure of the WSN, before reaching a state where the collected data are ready to be uploaded, the communication between the GW/CH and a sensor node is inevitably necessary. The mechanism of DPS enables both the sensor node and GW to predict the future value simultaneously based on the past measurements. In particular, the sensor node constantly compares the latest sensed data with the predicted data in each iteration, and only when the error exceeds a pre-defined threshold, a transmission is made. With the latest transmitted data, the GW/CH and sensor nodes, therefore, are capable of amending and updating the coefficients of the prediction model. Consequently, the entire process not only keeps track of the latest environmental changes, ensuring that the predicted data are fairly accurate by continuously updating the prediction model, it also reduces the total number and duration of data transmissions, which effectively conserves a significant amount of energy in long-term environmental monitoring applications.

A specific illustration of implementing the DPS is shown in Figure 3.2. Considering a scenario where a sensor node communicates with a CH, an *n* number of samples x(1), x(2), ..., x(n) are first collected by a sensor node and later transmitted to the CH. Meanwhile, a prediction model is initialized based on those samples. In the period of implementing DPS, both the sensor node and the CH constantly predict data, which are denoted as \hat{x} . Meanwhile, the sensor node also compares the latest predicted data with the latest sampled data in each iteration, and if the difference is tolerable, the sampled data will be discarded. On the other hand, if the difference exceeds a predefined threshold error $e_{threshold}$ (see x(n+3) and $\hat{x}(n+3)$), the sensor node will inform the CH by transmitting the sampled data, and the prediction model will be updated bilaterally.

The implementation of DPS requires a robust network protocol design, which means each time when a transmission is bound to happen, the next receiver (e.g., a sensor node, a CH, or the GW) could manage to receive the data. Failing to synchronize the data may result in failure of updating the coefficients of the prediction model, making the DPS implementation useless. The algorithm proposed in the present work assumes that there is no network failure and all the sensor nodes are placed within proper ranges.

	Initialize Prediction Model					Implement DPS			Update Model	Implement DPS			Update Model	Imple	Implement DPS	
СН	x(1)	x(2)	x(3)		x(n)	x̂(n+1)	x̂(n+2)	Â(n+3) ≜		∧ x(n+4)	̂x(n+5)	x̂(n+6)		∧̂(n+7)	∧ x(n+8)	
Sensor						x(n+1)	x(n+2)	x(n+3)		x(n+4)	x(n+5)	x(n+6)		x(n+7)	x(n+8)	
Node	x(1)	x(2)	x(3)		x(n)	x(n+1)	^ x(n+2)	^ х(n+3)		^(n+4)	^ х(n+5)	∧(n+6)		∧(n+7)	x(n+8)	

Time

Figure 3.2 The process of implementing DPS between a sensor node and the CH.

3.3 The LMS Algorithm

The LMS algorithm has been widely used in various environmental applications for time-series prediction due to its simplicity and robustness. It manifests itself as an autoregressive process when used, which only requires a low computational overhead and memory footprint in applications of WSN. Also, compared to some other filters such as the Kalman filter, the LMS algorithm does not necessarily need prior knowledge or nature (e.g., probability distribution) of the statistics of the environment, which thus efficiently reduces the need for significant training. The structure of the LMS algorithm is illustrated in Figure 3.3.

The training process of the LMS filter is done in a manner similar to the Back-Propagation Neural Network (BPNN), which constantly compares the predicted outputs with the desired outputs, and subsequently updates the weights of the hidden neurons, in each iteration. A detailed description of the algorithm is given next.

At any given time, an *N*-dimensional input signal vector X(n) collected by a sensor node is represented as

$$X(n) = [x(n-1), x(n-2), x(n-3), \dots, x(x-N)]$$
(3.1)

which comprises N readings from the past measurements. Hence, the output at time n, which is also the prediction value, is obtained as

$$y(n) = W(n)X^{T}(n)$$
(3.2)

where $W(n) = [w_1(n), w_2(n), w_3(n), ..., w_N(n)]$ is the weight vector. Typically, W(0) is set to 0, which could be the worst possible initializing condition for the algorithm [44].



Figure 3.3 Structure of the LMS algorithm.

With the output derived from (3.2), denote d(n) as the desired signal which the filter adapts to. Then the error e(n) between the desired signal and the prediction is given by

$$e(n) = d(n) - y(n) \tag{3.3}$$

Based on the computed error in (3.4), the weights can be updated as

$$W(n+1) = W(n) + \eta X(n)e(n)$$
 (3.4)

The present work utilizes a normalized LMS filter with a fixed step size 1, in which the weight coefficients are updated according to:

$$W(n+1) = W(n) + \frac{X(n)e(n)}{\|X(n)\|^2}$$
(3.5)

Practically, in environmental monitoring applications, the time-series measurements collected by a single node demonstrate the characteristics of temporal correlations. It is safe to say that the correlated measurements within a short sampling interval might be numerically similar. Also, recalling that a pre-defined threshold has been established in DPS (refer to Chapter 3.2 and Figure 3.2), the desired output is determined as:

$$d(n) = x(n) (e(n) > e_{threshold}) OR d(n) = y(n) (e(n) \le e_{threshold})$$
(3.6)

where the desired output d(n) is equal to the real sensed data x(n) while the prediction error is greater than the pre-defined error $e_{threshold}$, otherwise it equals the predicted output y(n).

3.4 The Long Short-Term Memory Recurrent Neural Network (LSTM RNN)

Constrained by the complexity of the original signal, the LMS filter may not provide the expected accurate prediction. Considering the fact that the signal of the ground-truth data might be decomposed as a combination of linear segments and nonlinear segments, it is expected that besides the LMS filter, a nonlinear model capturing the nonlinearity of the signal would enhance the prediction accuracy.

In such a context, the Neural Networks (NNs) are a convenient tool that have been widely studied and utilized in the past decades. The feedforward NNs are capable of providing satisfactory performance in capturing the nonlinear relations in given data, and also making predictions about them. Unfortunately, there are also shortcomings of feedforward NNs. One of the most challenging and obvious disadvantages of feedforward NNs is the lack of memory. That being said, when the inputs are time-series related, the forecasts should be somewhat correlated with the data in the past periods. While an NN is capable of mapping high-level nonlinear relationship, it treats each input time-independently, thus failing to yield a sound prediction.



However, the Recurrent Neural Networks (RNNs), compared to the traditional NNs, are capable of capturing the dependencies of long-range time-series sequences. That means, while processing the time-series sequences, the trained model acquires the correlation within the sequence, and the prediction is thereby influenced and determined by a long range of past inputs. Due to this superiority, RNNs have been widely utilized in various applications such as speech recognition, prescription systems, image processing, and natural language processing. Unfortunately, constrained by the inner structure of an RNN, the issue of gradient vanishing/exploding emerges, which necessitated the enhanced version LSTM RNN. A detailed description of LSTM is found in [45].

When dealing with time-series sequences, the LSTM normally outperforms RNNs and the Hidden Markov Model [25]. By avoiding the gradient vanishing/exploding problems, the LSTM

constantly updates the module state by selectively keeping or removing past inputs from a longterm range. Particularly, this benefits those environmental monitoring applications such as the rainfall prediction and water level prediction, where the collected data vary temporally, following a seasonal pattern.

In order to keep track of long sequences, the notion of memory cell is used. A memory cell contains three gates: input, output and forget. The three gates together control the inputs to flow in and out of the memory cell and decide how much information should be carried over from any arbitrary time. Figure 3.4 shows the structure of the LSTM cell.

It should be noted that the LSTM uses the cell state to iteratively store the information transferred among neurons. Hence, denoting the forget gate, input gate, output gate, and the cell state as f_t , i_t , o_t and C_t , respectively, the formulas for updating the module and carrying the information forward are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
(3.7)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.8}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.9}$$

$$C_{t} = \tanh(W_{c} \cdot [h_{t-1}, x_{t}] + b_{c})$$
(3.10)

$$C_t = f_t^* * C_{t-1} + i_t * C_t \tag{3.11}$$

$$h_t = o_t * \tanh(C_t) \tag{3.12}$$

48

where h_{t-1} is the output vector at the last time step, σ is the sigmoid function, $\{W_f, W_i, W_o, W_c\}$ $\in n^{x \ge n}$ is the weight matrix, and $\{b_f, b_i, b_o, b_c\} \in$ represent the bias matrix. In addition, (3.10) derives a new candidate value C_i , indicating by what scale the information needs to be retained and subsequently, being added to a new state in (3.11). Meanwhile, (3.9) and (3.12) collaboratively determines the output vector h_i at the current time step.

3.5 Hybrid DPS Based on LMS and LSTM

Since the LMS is capable of observing the linear features and making corresponding predictions, and also since the LSTM encompasses the advantage of making nonlinear predictions, it is envisaged that a Hybrid DPS relying on both methods could provide better predictions and further reduce the necessary transmission times in a WSN for environmental applications.

According to Equation (3.6), if the error goes beyond the pre-defined value, rather than directly transmitting the ground-truth data x(n), a beacon signal could be sent from the sensor node to CH, informing both sides to use the forecast value from LSTM at the current time step. Compared to a normal data packet that is used to update the prediction model, a beacon signal is typically of smaller size, which in return consumes lower energy while maintaining a reliable level of accuracy of the collected data. However, if both prediction models fail to provide a satisfactory prediction, which means the error is still intolerable, a packet containing the ground-truth data will be sent to

the CH and the prediction model will be updated bilaterally. Figure 3.5 illustrates the process of implementing LSTM with LMS.

Taking into account the illustration in Figure 3.2, if the difference exceeds a pre-defined threshold (again see x(n+3) and $\hat{x}(n+3)$), rather than directly transmitting x(n+3) and updating the prediction model, a forecast value h derived by LSTM at the current time step could be utilized as a substitute. If the error is acceptable at this time, a beacon signal consisting of a small packet is sent from the sensor node, informing that both sides should take h, as the valid prediction value. However, if the difference is intolerable between x(n+6) and $\hat{x}(n+6)$ and even the forecast value derived by LSTM at the corresponding time step could not yield a tolerable difference, then it becomes necessary to transmit the ground-truth value from the sensor node side and update the prediction model bilaterally. A pseudo code for implementing the Hybrid DPS is shown in Algorithm 3.1.

	Initia	lize Pr	edictio	on M	odel	Implement DPS			Vedate Implement DPS Update				Imple	Implement DPS		
СН	x(1)	x(2)	x(3)		x(n)	x̂(n+1)	𝔆(n+2)	x̂(n+3)		∧̂(n+4)	x̂(n+5)	x̂(n+6)		∧̂(n+7)	∧(n+8)	
							Beacc Signa	วท ป								
Sensor						x(n+1)	x(n+2)	x(n+3)		x(n+4)	x(n+5)	x(n+6)		x(n+7)	x(n+8)	
Node	x(1)	x(2)	x(3)		x(n)	x(n+1)	^ х(n+2)	∧ x(n+3)		∧(n+4)	∧ x(n+5)	∧ x(n+6)		∧(n+7)	∧(n+8)	

Time

Figure 3.5 Hybrid DPS indicating when the model needs updating.

Algorithm 3.1. A Hybrid DPS Based on LMS and LSTM consecutive Collect N samples initial as inputs: X(n) = [x(n-1), x(n-2), ..., x(x-N)], feed the inputs to both LMS and LSTM for model training **for** each time step n > N **do** 1. 2. measure the ground-truth sample x(n)3. estimate the prediction value by $y(n) = W(n)X^{T}(n)$ 4. estimate the prediction error by e(n) = d(n) - y(n)5. if $|e(n)| \leq e_{threshold}$ do 6. update X(n) = [y(n), x(n-2), ..., x(x-N+1)]update $W(n+1) = W(n) + \frac{X(n)e(n)}{\|X(n)\|^2}$ 7. 8. else do $e(n)' = d(n) - h_n$ 9. if $|e(N)'| \leq e_{threshold}$ do 10. update $X(n) = [h_n, x(n-2), ..., x(x-N+1)]$ 11. update $W(n+1) = W(n) + \frac{X(n)e(n)}{\|X(n)\|^2}$ 12. 13. else do 14. transmit the ground-truth sample x(n) to CH 15. update X(n) = [x(n), x(n-2), ..., x(x-N+1)]update $W(n+1) = W(n) + \frac{X(n)e(n)}{\|V(n)\|^2}$ 16. $\left\|X(n)\right\|^2$ 17. end

3.6 Simulation Study

3.6.1 Data Selection

For the computer simulation that is presented in this section, the data sets provided by the Intel Berkeley Research Lab are chosen from their website at db.csail.mit.edu/labdata/labdata.html [37]. There were 54 sensors in total deployed in their lab, successively collecting environmental data from February 28th to April 5th 2014. Various types of data including the humidity, temperature, light, and voltage values, sampled every 31 seconds, are available from this site. Researchers have widely depended on these data sets to validate their work. In the present work, the temperature data from Node 13 has been randomly selected as the data set. After removing some abnormal data and using interpolation to replace them, there are 25000 data samples in total in the used data set.

3.6.2 Experimental Setup and Parameter Determination

There is no specific rule to exactly determine the number of hidden layers and neurons in an NN, yet empirically-derived rules of thumb exist. Also, common experience could be followed for this purpose. Typically, two hidden layers are sufficient to handle a practical application since one hidden layer is adequate to represent any nonlinear function. A number greater than two may lead to over-training and also will unnecessarily cause the training process to be computationally expensive. Additionally, a rule of thumb indicates that the number of neurons in the hidden layer should be no more than the number of inputs but greater than the number of outputs. In the present simulation, the entire data set is divided into a training set with 21332 samples, a validation set

with 1123 samples, and a testing set with 2545 samples. The prediction is one-step ahead and carried out on a sliding-window basis. The used window size is 50, which means, each time, a vector containing 50 individual samples is fed into the network for predicting the next value. While the new prediction is appended at the very end of the vector, the oldest one will be removed. In this manner, repeatedly, the new vector is used for the next round of prediction. After many rounds of trial, the two hidden layer network structure provided a superior performance than with just one layer. However, in order to choose the most appropriate number of neurons in the first layer and the second hidden layer, the MSE and Mean Absolute Error (MAE) are utilized to measure the accuracy of the proposed algorithm on the test samples. The MSE and MAE are defined as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$$
(3.13)

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |x_i - \hat{x}_i|$$
(3.14)

where x_i denotes the ground-truth value in the test data set and \hat{x}_i denotes the predicted value. Table 3.1 lists the overall performance when different number of neurons is chosen for the first layer. Each value represents the average value obtained after 5 rounds of trial. By comparing the average MSE and MAE when the number of neurons in the second layer vary, it is found that when there are 50 neurons in the first layer, the average MSE and MAE reach their minimum values. Then, on fixing the number of neurons in first layer at 50, Table 3.2 presents the corresponding performance after 5 rounds of trial when different number of neurons are used in the second layer.

# of Neurons in 2 nd layer	# of Neurons in 1 st layer	50	45	40	35	30	25	20	15	10	5
	MSE	0.1723	0.1301	0.1447	0.1299	0.1997	0.1422	0.183	0.2151	0.2104	0.1698
50	MAE	0.0516	0.0352	0.0512	0.0449	0.0660	0.0498	0.0537	0.0727	0.0679	0.0538
40	MSE	0.1510	0.1584	0.1631	0.2118	0.1449	0.1985	0.1625	0.1617	0.1449	0.1201
40	MAE	0.0449	0.0413	0.0444	0.0664	0.0379	0.0601	0.0507	0.0451	0.0426	0.0364
30	MSE	0.1430	0.1644	0.2005	0.1745	0.1303	0.2036	0.1842	0.1270	0.1513	0.1704
50	MAE	0.0376	0.0443	0.0563	0.0633	0.0356	0.0633	0.0572	0.0427	0.0418	0.0511
20	MSE	0.1677	0.2183	0.1274	0.1484	0.1851	0.1230	0.0947	0.1565	0.1534	0.1870
20	MAE	0.0542	0.0666	0.0380	0.0411	0.0554	0.0334	0.0284	0.0491	0.0457	0.0654
10	MSE	0.1278	0.1533	0.1566	0.2199	0.1377	0.1843	0.1425	0.1408	0.1239	0.1923
10	MAE	0.0402	0.0477	0.0648	0.0711	0.0432	0.0534	0.0478	0.0531	0.0410	0.0600
Average	MSE	0.1524	0.1649	0.1585	0.1769	0.1596	0.1703	0.1535	0.1602	0.1568	0.1679
Avelage	MAE	0.1963	0.1970	0.2027	0.2299	0.2035	0.2173	0.1996	0.2203	0.2063	0.2187

TABLE 3.1 The number of neurons in the $1^{\mbox{\rm st}}$ layer and the corresponding error.
# of Neurons		50	45	40	35	30	25	20	15	10	5
Round 1	MSE	0.0542	0.0590	0.0245	0.0318	0.0328	0.0256	0.1002	0.0328	0.0329	0.1204
	MAE	0.1936	0.1772	0.0892	0.1274	0.1255	0.0930	0.2757	0.1129	0.1168	0.2208
Round 2	MSE	0.0339	0.0275	0.0292	0.0372	0.0320	0.0553	0.0727	0.0309	0.0640	0.0421
	MAE	0.1332	0.0930	0.1026	0.1302	0.1307	0.1793	0.2140	0.1191	0.1850	0.1444
Round 3	MSE	0.0470	0.0457	0.0457	0.0581	0.0312	0.0443	0.0462	0.0393	0.0307	0.0339
	MAE	0.1447	0.1680	0.1680	0.1941	0.1207	0.1647	0.1520	0.1215	0.0948	0.1154
Round 4	MSE	0.0826	0.0554	0.0554	0.0340	0.0419	0.0890	0.0273	0.0302	0.0289	0.0696
	MAE	0.2577	0.1987	0.1987	0.1274	0.1629	0.2028	0.1016	0.1092	0.1106	0.1727
Round 5	MSE	0.0404	0.0653	0.0653	0.0310	0.0502	0.0727	0.0244	0.0356	0.0446	0.0362
	MAE	0.1326	0.1986	0.1986	0.1015	0.1754	0.2023	0.0952	0.1233	0.1317	0.1099
Average	MSE	0.0516	0.0427	0.0449	0.0384	0.0376	0.0574	0.0542	0.0338	0.0402	0.0604
	MAE	0.1723	0.1453	0.1514	0.1361	0.1430	0.1684	0.1677	0.1172	0.1278	0.1526

TABLE 3.2 THE number of neurons in the 2^{ND} layer and the corresponding error.

It is observed that when the number of neurons equals 15, both MSE and MAE reach their lowest values. As a result, the LSTM designed for implementing the developed algorithm is comprised of 2 hidden layers with 50 neurons in the first hidden layer and 15 neurons in the second layer. The number of inputs is 50 and the number of outputs is 1. In addition, the dropout probability is set at 0.2. To avoid over-fitting (over-training), 50-epochs of training is conducted to determine the loss on the training data set and the validation set. Results are shown in Figure 3.6.



Figure 3.6 Model loss in 50 epochs of training.

Clearly, the validation loss continues to drop significantly until approximately the 9th epoch, after which the validation loss starts to fluctuate and eventually stabilizes at the level that is nearly the same as the level at the 9th epoch. So, it can be inferred that the model on the test data set starts to over-fit after the 9th epoch. Hence, 9 epochs are used for training the model.

In comparison with the LSTM, the training process of the LMS is relatively simple. Since a large number of samples is not needed for the training process, only 50 samples are taken for initialization. Also, the system input order is set at 5, which means in each iteration, 5 individual samples are fed into the filter. The pre-defined threshold is set at 0.2. It is suggested that the pre-defined threshold should vary depending on the specific characteristic of the parameter that is to be determined.

The simulation experiments using the developed algorithms are carried out in a PC with a 2.60 GHz Intel Xeon E5-2630 v2 CPU, 128GB RAM. The MATLAB R2017b and Python 3.6 with Keras using Tensorflow backend are the platforms that were used to conduct the simulation.

3.7 Simulation Results

With the pre-defined error set at 0.2, Figure 3.7 shows the simulation results for the estimated data, and compares with those for the original data, which consist of 2545 samples.

To further examine the performance of the proposed algorithm, Table 3.3 lists all the simulation results in terms of the prediction accuracy when the pre-defined error ranges from 0.1 to 0.6. Again, the MSE and MAE are utilized for measuring the accuracy.

Pre-defined error	0.1	0.2	0.3	0.4	0.5	0.6
MSE	0.0015	0.0090	0.0220	0.0395	0.0549	0.0750
MAE	0.0236	0.0686	0.1163	0.1605	0.1927	0.2251

 TABLE 3.3

 THE PREDICTION ACCURACY FOR DIFFERENT PRE-DEFINED ERROR VALUES.



Figure 3.7 Estimated data compared to the original data when $e_{threshold}=0.2$;

Depending on the difference in the pre-defined error values, the performance of the prediction varies. It should be noted that the selected pre-defined error determines the performance of the proposed algorithm. However, in practical applications, the pre-defined error is essentially a trade-off between the prediction accuracy and the energy consumption. Intuitively, a higher accuracy means a more frequent communication, which results in an increase of energy consumption. However, in terms of different data sets, there should be a general way of deriving the pre-defined error. Given an input denoted as X_i , the average, maximum and minimum values of this entire data set is identified as $AVG(X_i)$, $MAX(X_i)$, and $MIN(X_i)$, respectively. The pre-defined error $e_{threshold}$ is hence expressed as

$$e_{threshold} = MIN\left\{ \left| MAX(X_i) - AVG(X_i) \right| \times \eta\%, \left| MIN(X_i) - AVG(X_i) \right| \times \eta\% \right\}$$
(3.15)

where η is a user-defined parameter that indicates to what degree the error could be tolerated. In our case, the $e_{threshold}$ equals 0.2 when η % is set as 5%. In practical applications, it is suggested that the pre-defined threshold could be slightly adjusted either in favor of a higher prediction accuracy or energy conservation.

It is then believed that when the two time-series methods are combined together, the prediction accuracy is further improved when a proper value for the pre-defined error is selected. In particular, from Table 3.3 it is observed that when the pre-defined error is set to a number equal to or smaller than 0.3, both MSE and MAE are smaller than the prediction results obtained by LSTM alone. Also, as the pre-defined error increases, which means a relatively larger error would be tolerable, the proposed algorithm itself could smoothly handle the prediction process and meet the accuracy

requirements, and hence a fewer number of transmission would be needed. Table 3.4 lists all the energy consumption details for data transmission and sampling, which are measured in terms of the data size. The energy consumption E_{trans} for transmitting *l*-bit data through a distance *d* is defined as [46]:

$$E_{trans} = \begin{cases} l \times E_{elec} + l \times E_{fs} \times d^2, & d \le d_1 \\ l \times E_{elec} + l \times E_{amp} \times d^4, & d > d_1 \end{cases}$$
(3.16)

where E_{fs} is the energy consumed based on a free space model when the transmitting distance is within the distance limit d_l while E_{amp} stands for the energy consumed by an amplifier when the transmitting target is located outside of the distance limit d_l .

Parameter	Value
Basic energy consumed for transmission (E_{elec})	50 nJ/bit
Energy consumed based on a free space model (E_{fs})	10 pJ/(bit m ²)
Energy consumed by an amplifier (E_{amp})	0.0013 pJ/(bit m ⁴)
Energy consumed by sampling data (E_s)	0.5 nJ/(bit signal)

Table 3.4 Energy consumption details for data transmission and sampling.

It is clear that the energy consumed for data transmission is significantly larger than that consumed for sampling data in terms of the data size. Hence, to analyze the conserved energy by achieving a reduced transmission, the total number of transmissions is measured under different schemes. Figure 3.8 shows the necessary number of transmissions containing the data packets for updating the model and the times of sending beacon signals. In addition, as a comparison, the number of necessary transmissions while only either the LMS and the LSTM is utilized is provided. The blue bar shows the times of necessary transmission while solely LMS is used; the red bar indicates the times of necessary transmission while solely LSTM is utilized; the yellow bar represents for the number of necessary transmissions and the purple bar shows the times of sending beacon signals while using the developed Hybrid DPS.



Figure 3.8 Number of transmissions for different pre-defined error values.

It was demonstrated that the Hybrid DPS could reduce the number of network transmissions effectively. Compared to the case where only the LMS is used, the beacon signal, playing the role of a substitution, takes a reasonable amount of total transmissions as the pre-defined error increases. Beacon signals are also called advertising packets, which consist of small packets of data. However, the specific number of advertising packets should be based on the communication standard. In this manner, the energy consumed in network communication could be further conserved, while guaranteeing the accuracy of the collected data. For example, without a DPS being applied, there might be 2545 transmissions for sending all the measurements. However, when the Hybrid DPS is adopted, and $e_{threshold}$ is 0.2, less than one-third of the 2545 transmissions are needed plus the times of sending the beacon signals. Though it is known that the normal packet size depends on the specific application as well as the protocol, considering the fact that the beacon signals are the advertising packets, which are essentially smaller than the normal signals, the total energy consumption could still be reduced by more than a half. According to [27], for the ZigBee protocol, which is built based on the IEEE 802.15.14 standard, advertising packet is theoretically at least half the size (which is much smaller in reality) of a normal signal that contains the data payload. In that sense, the equation for computing the total packets needed to be sent is given by:

$$\begin{cases} P_T = P_N + \alpha P_B \\ \alpha = \frac{P_N}{P_B} \end{cases}$$
(3.17)

where P_T denotes the total packets, P_N denotes the packets of normal signals that contain the sensed data, P_B represents the packets of beacon signals, and α is the ratio of these two. Given that α is at least equal to 2, when $e_{threshold}$ is equal to 0.2, at least 62.29% of the total energy for data transmission could be saved. In the meantime, the MAE is only 0.0687, which makes the

prediction accuracy approximately 93% with respect to the original data. To further understand the performance of the proposed algorithm in terms of energy conservation, Figure 3.9 offers the comparison of energy expenditure based on Equation (3.17) and Figure 3.9. The total number of data packets, which is P_T that is referred to Equation (3.17), is used as a measurement. The curves show that the Hybrid DPS (LMS+LSTM) outperforms the two algorithms when each of them is utilized solel. It should be noted that in practical applications, the value of α could be greater than 2, which means the Hybrid DPS might have an even better performance than that indicated in the figure. Moreover, there exists a trade-off between the prediction accuracy and the energy consumption. Table 3.5 presents the maximum absolute prediction error among the prediction results and Figure 3.10 also indicates the relationship among the threshold, accuracy and the level of energy conservation, so it could be adjustable depending on the specific applications and the anticipated accuracy as well as the remaining network energy.

Pre-defined error	0.1	0.2	0.3	0.4	0.5	0.6
Hybrid DPS	0.0998	0.1996	0.2995	0.3996	0.4994	0.5985
LMS only	0.0999	0.1997	0.2995	0.3997	0.4994	0.5993
LSTM only	0.0999	0.1998	0.2999	0.3987	0.4999	0.5941

TABLE 3.5 MAXIMUM ABSOLUTE PREDICTION ERROR (IN ° CELSIUS) FOR DIFFERENT PRE-DEFINED ERROR VALUES.



Figure 3.9 Data packets sent in terms of different pre-defined error values.



Figure 3.10 Trade-off between accuracy and energy consumption for different threshold values.

Noting that a reduced communication level would help save energy in the WSN, selecting a proper number of sensor nodes could also be important in the environmental monitoring applications. Particularly, it is vital when it comes to covering the entire monitored area with the least allowable number of sensor nodes. Chapter 4 presents a methodology for estimating a proper number of sensor nodes.

Chapter 4: Optimal Estimation of Sensor Nodes

The selection of the number of deployed sensor nodes in the region of interest (ROI) is a nontrivial problem in environmental monitoring applications where a WSN is used. Particularly, a large number of sensor nodes indicates a heavier overhead within the network and consequently, more energy will be consumed in the tasks of sensing, communication and possible sensor node relocation. However, if the number of sensor nodes is not sufficiently large, some key areas within the ROI may not be fully or properly covered and thus some key information might not be collected. To decide an optimal number of sensor nodes, the notion of a sensing model has to be introduced. This is done next

4.1 Sensing Models

Given that the ROI is considered in a two dimensional plane in the considered application (i.e., water quality monitoring), two types of sensing models are widely utilized for WSN applications. They are the binary disk sensing model and the probabilistic sensing model. Within any ROI, denote a random sensor node as n_i , a random point as p, and the probability of detecting a target at point p as $\rho(n_i,p)$. Then the binary disk sensing model is formulated as:

$$\rho(n_i, p) = \begin{cases} 0, & \text{if } d(n_i, p) > R_s \\ 1, & \text{otherwise} \end{cases}$$
(4.1)

where $d(n_i,p)$ represents the Euclidean distance between the sensor node and the point p, while R_s stands for the sensing radius of the sensor node n_i . It is quite clear that if a random point falls outside the sensing radius, it would not be detected by the sensor node. Otherwise, detection of

the point would be guaranteed. However, for the probabilistic model, the detection of an event follows a probabilistic distribution pattern, which is defined as:

$$\rho(n_i, p) = \begin{cases}
0, & \text{if } d(n_i, p) > R_s \\
e^{-\alpha d(n_i, p)^{\beta}}, & \text{if } R_s > d(n_i, p) > R_c \\
1, & \text{otherwise}
\end{cases}$$
(4.2)

where α and β are the parameters determined by the inherent characteristics of the sensor node, which also indicate the difference in sensing capability for different types of sensor nodes. R_c represents an alternative radius, within which the probability of detecting an event is 1. Figure 4.1 shows two types of sensing probabilities in the sensing models.



Figure 4.1 (a) Binary sensing disk model; (b) Probabilistic sensing model

This thesis considers the binary sensing disk model for further study. Regarding each sensor node in the ROI, with only a single sensing radius that contributes to the total sensing coverage, the coverage problem hence becomes straightforward. A probabilistic sensing model for the coverage problem will be considered in the future work.

4.2 Number of Sensor Nodes

Deciding an optimal number of sensor nodes that meets the monitoring requirements is also critical from the perspective of the data sampling efficiency as well as the energy consumption. However, determining an optimal number of sensor nodes in the ROI is NP-hard [47]. Hence a few heuristic methodologies have been proposed to find an optimal number for monitoring applications in which the WSN/MWSN are utilized. However, most of them rely on a probabilistic model and a prior sampling knowledge of the ROI is also required. As the research presented in [48], with a prior sampling knowledge of the ROI, the sensor nodes are most likely placed in "information rich" locations, where the sensor node can collect massive amounts of useful data or be able to detect specific events at high probability in the ROI. Likewise, in [49], the sensor nodes are dynamically placed one after the other based on the probability of being able to communicate with other sensor nodes. The placement stops when the ROI is mostly covered by the sensor nodes while the connectivity of the sensor nodes is guaranteed. However, for an unknown environment where no prior knowledge is available, a general way of estimating a proper number of sensor nodes would be needed. Particularly, for a coverage problem where the energy consumption is also a concern, there is no doubt that a large number of sensor nodes could cover the entire monitoring area sufficiently, but a heavier network overload is expected as a consequence.

Hence, it is desired to estimate a suitable minimum number of sensor nodes that are approximately adequate to cover most of the monitored area and as a result, excessive energy consumption can

68

be avoided in view of the absence of massive node relocation and communication. To fulfill this requirement, a solution is provided next, with an example.



Figure 4.2 First step of deploying sensor nodes to cover most space of a rectangular area.

Despite the fact that the minimum disk covering problem is NP-complete [50], the practical monitoring applications are different. For instance, given an ROI whose length D_1 is equal to 8 times the sensing radius R_s and width D_2 is 6 times the sensing radius, the first step of covering most of the ROI with least number of sensor nodes would be to deploy 12 sensor nodes without overlap, as shown in Figure 4.2. From Figure 4.3, it is noticed that there are still quite a few areas that are not covered, such as S_1 , S_2 and S_3 , which calls for more sensor nodes.



Figure 4.3 Areas not covered by sensor nodes after the first round of deployment.

Clearly, the total area of S_1 is exactly twice that of S_2 , while S_2 is twice that of S_3 . It is worth mentioning that in practical monitoring applications, the shape of the monitored area could potentially be irregular, which indicates that not all the points inside the area could be covered. Hence, if the entire area is partitioned into a number of sub-areas, and due to the possible irregularity of the area, it is believed that a suitable number for the sensor nodes can be derived by partitioning the entire area into sub-areas and subsequently recombining them as a whole. That being said, if one sensor node is capable of covering S_1 , for every other two areas that are of the same size as S_2 , one more sensor is needed. The same applies for every other four areas like S_3 . Hence, for a monitored area as shown in Figure 4.3, besides the initially deployed sensor nodes, 12 more sensor nodes are required to basically cover the entire monitored area. To generalize, in order to approximately cover the entire ROI, a reasonable number of sensor nodes may be expressed as:

$$N_{\max} = \frac{D_1}{2R_s} \cdot \frac{D_2}{2R_s} \cdot 2$$

$$= \frac{D_1 D_2}{2R_s^2}$$
(4.3)

To justify Equation (4.3), it can also be written as $N_{\text{max}} \cdot \pi R_s^2 = \frac{\pi}{2} \cdot D_1 D_2$. Then, the total sensing

coverage of N_{max} sensor nodes, equals $\frac{\pi}{2} \cdot D_1 D_2$, which is greater than $D_1 D_2$, namely the total

area of the ROI. Moreover, since $\frac{D_1}{2R_s}$ and $\frac{D_2}{2R_s}$ sometimes are not divisible, the equation should

be rewritten, more accurately, as:

$$N_{\max} = \left\lceil \frac{D_1}{2R_s} \right\rceil \cdot \left\lceil \frac{D_2}{2R_s} \right\rceil \cdot 2$$
(4.4)

With the developed methodology for selecting a proper number for nodes, Chapter 5 demonstrates an energy-efficient scheme for sensor node relocation, which aims to save as much energy as possible with least node movement, in order to maximize the total sensing coverage.

Chapter 5: MWSN Lifetime and Maximum Sensing Coverage

A large geographical area could possibly be monitored by using a sufficiently large number of sensor nodes in order to collect all the information within the environment. However, extra energy will also be consumed due to the increased number of sensor nodes. Chapter 4 provides a solution for covering the monitored area effectively while using a fewer number of sensor nodes. Continuing in this direction, the present chapter aims to maximize the total sensing coverage with an estimated number of sensor nodes. A sensor node relocation scheme associated with this approach is presented in this Chapter.

Considering various types of monitoring applications, the sensor nodes might have to be located in hazardous environments that may even be inaccessible. Hence, efficiently making use of the limited energy and prolonging the lifetime of the entire sensor network might be a major concern that needs to be properly addressed. In this regard, careful node placement could be an effective solution for energy conservation and allocation. Incorporating proper node placement is also advantageous in situations where the sensor nodes are mobile, as in the present application of water quality monitoring. This means the mobile sensor nodes could be relocated with respect to the dynamic environmental changes or the remaining energy of the nodes.

The main incentives for studying the sensor node placement in mobile sensor networks can be summarized as follows: 1) maximizing the lifespan of the sensor network; 2) ensuring maximum coverage of the region of interest (ROI); 3) maximizing the connectivity among sensor nodes; 4) minimizing the energy wastage. Also, considering the fact that the sensor nodes within the ROI might be faulty or malfunctioning, especially when a large number of sensor nodes are deployed in a hazardous situation, the distribution of sensor nodes will become uneven and consequently, the energy of some sensor nodes might deplete faster than that of others. It is then desirable to relocate the sensor nodes and reorganize the structure of the entire sensor network, which will help achieve a fair balance and prolong the network life.

5.1 Assumptions

Pertinent assumptions are made now, before delving further into the proposed algorithm.

Assumption 1: Every sensor node knows the location of every other sensor node through some technique such as a Global Positioning System (GPS) [51].

Assumption 2: Each sensor node has a disk of sensing range R_s and a communication range R_c , which are the same for an individual node.

Assumption 3: There are no obstacles in the ROI that could obstruct the node placement. This assumption is made for the sake of simplicity, but in future work, more complicated scenario that include static and dynamic obstacles would be incorporated.

5.2 Hole detection strategy based on Voronoi diagram

The VD is a widely used tool in computer graphics. It is capable of partitioning the entire ROI by generating convex polygons within the field. For each convex polygon c_i , with only one node n_i residing inside the polygon, their spatial collaborative relationship can be defined as:

$$c_{i} = \left\{ x \in {}^{2} : d(x, n_{i}) \le d(x, n_{j}), j \in \{1, \dots, N\}, i \ne j \right\}$$

$$(5.1)$$

where n_i denotes the neighbor node in an adjacent polygon, and $d(x,n_i)$ represents the Euclidean distance between the sensor node n_i and any random point x within the polygon c_i . This entails that any given point is closer to the node in the same polygon than to those in other polygons. In addition, this thesis considers a binary sensing model [52][53][54] under the condition that $R_c \ge 2R_s$, which is a necessity for maintaining full sensing coverage and connectivity in any given sensing field [55]. With a sensing range R_s , each node n_i is able to either fully or partially cover its corresponding polygon, depending on the shape of the partitioned polygon and the initial location of the node. Hence, the detection and coverage problem of "holes" in the entire ROI could be converted into studying the spatial correlations of the polygon c_i and the sensor nodes n_i . As illustrated in Figure 5.1, each node in the ROI takes its own responsibility to monitor its surrounding area. However, since there might be an overlapping area as well as an uncovered area in a convex polygon, the node may have to be relocated for better coverage and improved usage efficiency, taking account of the node-to-node connectivity, and also the moving distance.



Figure 5.1 The partitioned Voronoi polygons with nodes residing inside.

As for the hole detection strategy, for each single polygon, the corresponding node n_i should check its distance to every vertex V_k , that is:

$$d(n_i, v_k) = \|C_{n_i}(x, y) - C_{v_k}(x, y)\|$$
 (5.2)

where $d(n_i, v_k)$ represents the Euclidean distance between the node n_i and the vertex V_k while C_{ni} and C_{Vk} are their corresponding coordinates. Then, for every node n_i , the corresponding polygon is assumed to be not fully covered iff:

$$\forall d(n_i, v_k) \in {}^2, k \in \{1, \dots, N\}, \exists d(n_i, v_k) < R_s \quad \P \quad (5.3)$$

When one or more holes are detected, due to the characteristics of the convex polygon, it is reasonable that the node should move towards the vertex that is farthest to the node's current location, which could achieve an enhanced coverage rate in the ROI. The farthest distance for sensor n_i is given by:

$$d_{max}(i) = max \left\{ d\left(n_{i}, v_{k}\right) \right\}, \ k \in \{1, \dots, N\}$$

75

5.3 Hole Healing Strategy

According to the work reported in [56], the optimal position for a node should be somewhere between its initial location and the geometric centroid (GC) of the Voronoi polygon (VP). That means while aiming to move and heal the detected hole within the polygon, there should be a constraint for the node's movement. However, it should be noted that both the energy consumption for movement and the node-to-node connectivity should be taken into consideration in this scheme. Hence, a local safe area is built to ensure the connectivity while the node is moving. As illustrated in Figure 5.2, when the node n1 is moving towards GC, there should be at least one node whose distance to node n1 is not farther than Rc. In this manner, the restricted points f1 and f2 would be the last positions where n1 is able to communicate with n3 and n2, respectively. Despite the fact that locating at the point GC might possibly ensure a better coverage rate, the node n1 still has to stop at point f2 to maintain the connectivity within the entire network.



Figure 5.2 Node's movement for healing the detected hole.

Compared to the energy consumed for conducting sensing and communication tasks, the energy used for node movement is much more significant. Thus, as a trade-off for a better coverage rate, it is nontrivial to carefully decide when to stop the moving node for an optimal rearrangement. It should be pointed out that, no matter where the GC and restricted points lie in a VP, the coverage rate might stay the same or even decrease as the node moves. As can be seen in Figure 5.3, compared to the original location of n_1 , the same node moving towards vertex V_4 would provide a superior coverage rate at point n_1 ' than that at points n_1 and n_1 ''. Therefore, a threshold should be set to decide when to stop the node. In this thesis, it is assumed that when the coverage rate no longer increases more than 5% in the given iteration, the node stops moving in its corresponding polygon.



Figure 5.3 Changes of the covered area when node n_1 moves.

To implement the proposed scheme for the process of hole detection and healing, a pseudo code is provided as follows:

Algorithm 5.1

- 1. Randomly deploy *N* sensors in the ROI
- 2. Inputs: coordinates of all the deployed nodes n_i
- 3. Build up VD relying on $n_i(x,y)$ in the ROI
- 4. repeat
- 5. for each VP P(i), while $i \leq N$ do
- 6. calculate: $d(n_i, v_k) = \sqrt{(x_{n_i} x_{v_k})^2 + (y_{n_i} y_{v_k})^2}$
- 7. calculate: $d_{max}(i) = \max\{d(n_i, v_k)\}$ // determine the farthest vertex
- 8. calculate GC based on the coordinate of each vertex
- 9. calculate the movement direction based on GC and the farthest vertex
- 10. based on the communication range and the position of neighbor nodes, calculate the approachable movement area and restricted points in terms of the movement direction
- 11. calculate $d_{max}(n_i, f_i) = max \{d(n_i, f_i)\} //$ determine the farthest restricted point
- 12. **if** $d(n_i,GC) > d_{max}(n_i,f_j)$ // distinguish the spatial relationship between GC and the farthest restricted point
- 13. while $d_{move}(n_i) \leq d_{max}(n_i, f_i)$ 14. move node n_i towards f_i 15 calculate the movement distance as $d_{move}(n_i)$ 16. if $R_{cov} < 5\%$ then break // coverage increasing rate check 17. end if 18. end while 19. else // the else case entails $d(n_i, GC) \le d_{max}(n_i, f_i)$ while $d_{move}(n_i) < d(n_i, GC)$ 20. move node n_i towards GC 21. calculate the movement distance as $d_{move}(n_i)$ 22. if $R_{cov} < 5\%$ then break // coverage increasing rate check 23. end if 24. end while 25. end if 26. if P(i)!=NULL then i++, check next polygon P(i)27. end for 28. Outputs: coordinates of all the deployed nodes after movement

5.4 Illustrative Experiment

To demonstrate and verify the efficiency of the proposed scheme for hole detection and healing, an illustrative simulation experiment is presented in this section. Specifically, 30 sensor nodes are chosen to be randomly deployed in a $50x50m^2$ ROI with the R_s =6m and R_c =12m. The node moves through 0.5m in each iteration. Due to the stopping criterion that has been set up beforehand, no perturbation of the nodes will occur, which saves extra energy.

Figure 5.4(a) presents the initial locations of the nodes that are randomly deployed in the ROI while Figure 5.4(b) shows the locations of nodes after using the proposed scheme for repositioning. It is clear that the sensing coverage rate has been significantly increased, notably from 68.9% to 92.6%. Figure 5.5 also shows the guaranteed connectivity after the node repositioning. Clearly, all the sensor nodes within this ROI are connected to at least one surrounding node, which makes sure that all the collected data from each single node could be properly transmitted and uploaded through some routing technique [57]-[61].



Figure 5.4 (a) Randomly deployed nodes before repositioning; (b) Deployed nodes after repositioning.

Moving the nodes also costs significant energy compared to what is consumed for the tasks of data sensing and transmission [46][58]. To compare the proposed algorithm with the existing research that handles the same problem of sensing coverage optimization with regard to the energy consumption, the realistic parameters with values [59] involved in the relocation process are listed in Table 5.1.



Figure 5.5 Guaranteed node-to-node connectivity after repositioning.

Parameter	Value
Sensing range (R_s)	6 m
Communication range (R_c)	12 m
Data packet size	1000 bytes
Beacon packet size	45 bytes
Energy for node movement (E_m)	2 J/m
Energy for data processing (E_p)	5 nJ/(bit signal)
Transmitter electronics (E_t)	50 nJ/(bit signal)
Receiver electronics (E_r)	50 nJ/(bit signal)

Table 5.1 Realistic parameters with values.

Undoubtedly, data processing and computing will consume some energy. Also, data transmission among sensor nodes for updating the location information is necessary, which also consumes some energy. However, compared to the energy consumed for the above activities, the energy utilized for the movement of the nodes is so significant that the other types of energy consumption could be neglected. Likewise, as discussed in [58], moving a sensor node one meter consumes approximately a similar amount of energy for transmitting 300 messages. Hence, the present thesis mainly takes the travelled distance as the metric to indicate the efficiency of the proposed algorithm. Consequently, the total travelled distance is adopted to represent the efficiency of the developed algorithm compared to some existing algorithms.

Figure 5.6 depicts the total distance that all the nodes have travelled during the repositioning process.

To achieve the same sensing coverage rate, which is set at 90% in the present simulation, different numbers of nodes have been adopted for 10 rounds of simulation. The larger the total distance needed for achieving the same sensing coverage rate, the more energy is correspondingly consumed. As can be observed from the figure, the developed algorithm GCVD outperforms two traditional methods VFA in [27] and VOR in [29] concerning the energy consumption. It should be noted that when the number of nodes increases, the location adjustment to each node is still needed under the schemes of VFA and VOR, which consequently will cost extra energy for the node perturbation. However, the developed scheme GCVD in the present work does not necessarily have to move a node if its corresponding polygon has been well covered. Thus, especially when a larger number of nodes are deployed in the ROI, the percentage of nodes that need repositioning would be relatively lower in the developed scheme than for VFA and VOR.



Figure 5.6 Total distance travelled on average when different number of nodes are chosen.

This chapter presented a novel approach based on Voronoi diagram and its geometric center to determine the optimal sensor node location for sensing coverage maximization while ensuring the connectivity among sensor nodes with least node movement. Compared to the existing algorithms that handle the same problem, the proposed algorithm GCVD is found to be superior in relation to the energy conservation. Nevertheless, it is envisaged that the proposed scheme could be further enhanced, referring to the simulation results in Figure 5.4. For instance, some nodes such as node 12 and 13, could be selectively switched off since their sensing area could be covered by their neighbor nodes. Additionally, a new VD may be generated for relocating the nodes again, in order to further enhance the sensing coverage rate. This makes sense because there would always be a trade-off between the sensing coverage rate and the energy consumption for moving the nodes. These issues will remain as future work.

Chapter 6: Conclusion and Future work

6.1 Main Contributions and Significance

This thesis developed and evaluated new power management schemes for energy efficient sensor node utilization in a mobile sensor network for environmental monitoring activities, particularly water quality monitoring. The developed schemes have been designed from the perspective of single sensor node level, node-to-node level, and the entire network level. Through computer simulation using practical sensory data, it was demonstrated and verified that all the proposed schemes were highly efficient in conserving energy as well as maintaining the data accuracy. Thus, it is believed that all the proposed schemes are feasible for practical application in an effective manner, depending on the specific characteristics of the monitoring application.

The developed DDASA focuses on the power consumption issue on the single sensor node level. In this scheme, by constantly detecting if there is a sudden environmental change based on the newly sampled data, the sampling frequency is dynamically changed from the viewpoint of saving energy. It has been demonstrated and validated as well that the data accuracy will not be undermined while energy for the sensing activities will be further conserved. It was verified that this algorithm was robust for different types of parameter sampling, and could effectively conserve energy by using a satisfactory reconstructed signal. Compared with some existing adaptive sampling algorithms, which are battery-state driven, there are justifiable occasions where the sampling frequency is based on the real-time sampled data, especially when the fluctuation of the environmental parameters, is of significant interest. This commonly occurs in spatiotemporal monitoring of the quality of natural bodies of water, which is the specific practical application that 85 is addressed in the present dissertation. Additionally, it was shown that, by dynamically changing the sampling frequency according to the newly sampled data, the proposed DDASA would outperform a traditional adaptive sampling algorithm with respect to the accuracy of the reconstructed signal and the energy conservation. Thus, the goal of prolonging the life of the nodes has been achieved with the proposed approach, which is particularly advantageous in water quality monitoring applications in remote and hazardous locations.

Moreover, a hybrid dual prediction scheme was developed in the present dissertation, which was aimed to make time-series prediction in environmental applications where a wireless sensor network would be utilized. In the two-way communication between a normal sensor node and a CH, the developed algorithm effectively reduced the number of necessary transmissions by forecasting the future data, bilaterally. Meanwhile, the prediction accuracy could be maintained according to a pre-defined error threshold.

In addition, with regard to the network failure, since the proposed algorithm is capable of achieving a significantly reduced communication effort quantified by the total times of transmission, resending another data packet if network failure happens would not particularly burden or overload the network. Thus, for simplicity, this research assumed that there was no network failure.

Lastly, this thesis developed a novel approach based on Voronoi diagram and its geometric center, to determine the optimal location for sensing coverage maximization while ensuring the connectivity among sensor nodes, under least node movement. Compared to the existing algorithms that address the same problem, the proposed algorithm GCVD outperforms them with respect to the energy conservation. However, it is envisaged that the proposed scheme could be further enhanced, referring to the simulation results in Figure 5.4. For instance, some nodes such as node 12 and 13, could be selectively switched off since their sensing area could be covered by their neighbor nodes.

6.2 Possible Future Work

For the proposed algorithm GCVD, a new VD may be generated for relocating the nodes again, in order to further enhance the sensing coverage rate. This makes sense because there would always be a trade-off between the sensing coverage and energy consumption for moving the nodes.

Additionally, a more complicated scenario where obstacles get in the way of the moving nodes can be studied. The moving nodes may either stop early or detour during the relocation process. In this regard, the node-to-node connectivity might be temporarily lost and more energy might be consumed for detouring. Thus, for the same optimization problem, a more complex case with obstacles being considered is also worthy to be investigated.

Bibliography

- X. Liu, "Atypical Hierarchical Routing Protocols for Wireless Sensor Networks : A Review," *IEEE Sens. J.*, vol. 15, no. 10, pp. 5372–5383, 2015.
- [2] S. Kandukuri, J. Lebreton, R. Lorion, N. Murad, and J. Daniel Lan-Sun-Luk, "Energyefficient data aggregation techniques for exploiting spatio-temporal correlations in wireless sensor networks," *Wirel. Telecommun. Symp.*, vol. 2016-May, 2016.
- [3] B. Akan, I. F. Akyildiz, and M. C. Vuran, "Spatio-temporal correlation: theory and applications for wireless sensor networks," vol. 45, pp. 245–259, 2004.
- [4] J. Chen *et al.*, "Rapidly-Exploring Tree With Linear Reduction : A Near-Optimal Approach for Spatiotemporal Sensor Deployment in Aquatic Fields Using Minimal Sensor Nodes," *IEEE Sens. J.*, vol. 18, no. 24, pp. 10225–10239, 2018.
- [5] J. Guo and H. Jafarkhani, "Sensor Deployment With Limited Communication Range in Homogeneous and Heterogeneous Wireless Sensor Networks," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 10, pp. 6771–6784, 2016.
- [6] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Comput. Networks*, vol. 67, pp. 104–122, 2014.
- [7] S. Lin, "ATPC : Adaptive Transmission Power Control for Wireless Sensor Networks," vol.
 12, no. 1, pp. 1–31, 2016.
- [8] C. Lee and J. Lee, "Harvesting and Energy Aware Adaptive Sampling Algorithm for Guaranteeing Self-sustainability in Wireless Sensor Networks," 2017 Int. Conf. Inf. Netw., pp. 57–62, 2017.
- [9] N. Saeed, M. Murad, M. Nawaz, and M. Irum, "Survey on Single Path and Multipath Energy Efficient Routing Protocols for Wireless Sensor Networks," pp. 1–11, 2017.

- [10] W. Drira, K. Ahn, H. Rakha, and F. Filali, "Development and Testing of a 3G/LTE Adaptive Data Collection System in Vehicular Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 240–249, 2016.
- [11] R. Prabha, M. V. Ramesh, S. Member, V. P. Rangan, P. V Ushakumari, and T. Hemalatha,"Landslide Monitoring Systems," vol. 17, no. 18, pp. 6006–6018, 2017.
- [12] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, "An Adaptive Sampling Algorithm for Effective Energy Management in Wireless Sensor Networks With Energy-Hungry Sensors," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 1–23, 2010.
- [13] V. Raghunathan, S. Generiwal, and M. Srivastava, "Emerging Techniques for Long Lived Wireless Sensor Networks," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 108–114, 2006.
- [14] S. Santini and K. Römer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," *Proc. 3rd Int. Conf. Networked Sens. Syst. (INSS 2006)*, pp. 29–36, 2006.
- [15] G. M. Dias, B. Bellalta, and S. Oechsner, "The impact of dual prediction schemes on the reduction of the number of transmissions in sensor networks," *Comput. Commun.*, vol. 112, pp. 58–72, 2017.
- [16] C. Guestrin, P. Bodi, R. Thibau, M. Paski, and S. Madde, "Distributed regression," Proc. third Int. Symp. Inf. Process. Sens. networks - IPSN'04, p. 1, 2004.
- B. Stojkoska, D. Solev, and D. Davcev, "Data Prediction in WSN using Variable Step Size
 LMS Algorithm," *Sensorcomm 2011*, no. c, pp. 191–196, 2011.
- [18] H. Wei, J. He, and J. Tan, "Layered hidden Markov models for real-time daily activity monitoring using body sensor networks," *Knowl. Inf. Syst.*, vol. 29, no. 2, pp. 479–494, 2011.
- [19] G. Wei, Y. Ling, B. Guo, B. Xiao, and A. V. Vasilakos, "Prediction-based data aggregation

in wireless sensor networks: Combining grey model and Kalman Filter," *Comput. Commun.*, vol. 34, no. 6, pp. 793–802, 2011.

- [20] J. P. Hermias and J. C. N. Monje, "Short-Term Stochastic Load Forecasting Using Autoregressive Integrated Moving Average Models and Hidden Markov Model," pp. 131– 137, 2017.
- [21] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [22] R. A. Moghadam and M. Keshmirpour, "Hybrid ARIMA and Neural Network Model for Measurement Estimation in Energy-Efficient," pp. 35–48.
- [23] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to Diagnose with LSTM Recurrent Neural Networks," pp. 1–18, 2015.
- [24] Y. Wang, J. Zhou, K. Chen, Y. Wang, and L. Liu, "Water quality prediction method based on LSTM neural network," 2017 12th Int. Conf. Intell. Syst. Knowl. Eng., pp. 1–5, 2017.
- [25] X. Li, S. Wu, and L. Wang, "Blood Pressure Prediction via Recurrent Models with Contextual Layer," Proc. 26th Int. Conf. World Wide Web - WWW '17, pp. 685–693, 2017.
- [26] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," *Proc. - IEEE INFOCOM*, vol. 4, no. C, pp. 2302–2312, 2005.
- [27] X. Yu, "A novel virtual force approach for node deployment in wireless sensor network,"
 2012 IEEE 8th Int. Conf. Distrib. Comput. Sens. Syst., no. 2011, pp. 359–363, 2012.
- [28] A. Tripathi, H. P. Gupta, T. Dutta, R. Mishra, K. K. Shukla, and S. Jit, "Coverage and Connectivity in WSNs: A Survey, Research Issues and Challenges," *IEEE Access*, vol. 6, pp. 26971–26992, 2018.
- [29] N. Heo and P. K. Varshney, "Energy-efficient deployment of Intelligent Mobile sensor
networks," *IEEE Trans. Syst. Man Cybern. Part A Syst. Humans*, vol. 35, no. 1, pp. 78–92, 2005.

- [30] N. Xia, S. Member, C. Wang, S. Member, Y. Yu, and I. Member, "A Path Forming Method for Water Surface Mobile Sink Using Voronoi Diagram and Dominating Set," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7608–7619, 2018.
- [31] C. K. Liang, M. C. He, and C. H. Tsai, "Movement assisted sensor deployment in directional sensor networks," *Proc. - 2010 6th Int. Conf. Mob. Ad-hoc Sens. Networks, MSN 2010*, pp. 226–230, 2010.
- [32] A. Ray and D. De, "Performance evaluation of tree based data aggregation for real time indoor environment monitoring using wireless sensor network," *Microsyst. Technol.*, vol. 23, no. 9, pp. 4307–4318, 2017.
- [33] X. Liu, J. Cao, S. Lai, C. Yang, and Y. L. Xu, "Energy Efficient Clustering for WSN-based Structural Health Monitoring," 2011 Proc. IEEE INFOCOM, pp. 2768–2776, 2011.
- [34] C. Engineering, "MONITORING CLIMATIC CONDITIONS USING WIRELESS SENSOR," vol. 3, no. 1, pp. 179–184, 2017.
- [35] M. Vo, T. T. T. Nghi, V. Tran, L. Mai, and C. Le, "Wireless Sensor Network for Real Time Healthcare Monitoring :," pp. 87–91, 2015.
- [36] B. Srbinovski, M. Magno, F. Edwards-Murphy, V. Pakrashi, and E. Popovici, "An energy aware adaptive sampling algorithm for energy harvesting WSN with energy hungry sensors," *Sensors (Switzerland)*, vol. 16, no. 4, pp. 1–19, 2016.
- [37] and R. T. P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, "Intel Lab Data," Web page, Intel, 2004. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html.
- [38] C. Almhana, V. Choulakian, and J. Almhana, "An efficient approach for data transmission

in power-constrained wireless sensor network," IEEE Int. Conf. Commun., 2017.

- [39] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J. H. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Commun.*, vol. 14, no. 7, pp. 127–136, 2017.
- [40] R. L. Ali, S. A. Khan, A. Ali, Anis-Ur-Rehman, and S. A. Malik, "A robust least mean square algorithm for adaptive array signal processing," *Wirel. Pers. Commun.*, vol. 68, no. 4, pp. 1449–1461, 2013.
- [41] M. Razzaq, D. Devi Ningombam, and S. Shin, "Energy efficient K-means clustering-based routing protocol for WSN using optimal packet size," *Int. Conf. Inf. Netw.*, vol. 2018-Janua, no. 1, pp. 632–635, 2018.
- [42] A. Al-Baz and A. El-Sayed, "A new algorithm for cluster head selection in LEACH protocol for wireless sensor networks," *Int. J. Commun. Syst.*, vol. 31, no. 1, pp. 1–13, 2018.
- [43] K. Cengiz and T. Dag, "Energy Aware Multi-Hop Routing Protocol for WSNs," IEEE Access, vol. 6, 2017.
- [44] S. Haykin, *Neural Networks and Learning Machines*, vol. 3. 2008.
- [45] R. L. Anderson, "LSTM: A search space odyssey," J. Am. Stat. Assoc., vol. 48, no. 264, p.
 789, 1953.
- [46] L. Malathi, R. K. Gnanamurthy, and K. Chandrasekaran, "Energy efficient data collection through hybrid unequal clustering for wireless sensor networks," *Comput. Electr. Eng.*, vol. 48, pp. 358–370, 2015.
- [47] X. Xu and W. Liang, "Placing optimal number of sinks in sensor networks for network lifetime maximization," *IEEE Int. Conf. Commun.*, 2011.
- [48] C. C. Castello, J. Fan, A. Davari, and R. X. Chen, "Optimal sensor placement strategy for

environmental monitoring using wireless sensor networks," *Proc. Annu. Southeast. Symp. Syst. Theory*, pp. 275–279, 2010.

- [49] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," *IEEE Wirel. Commun. Netw. Conf. WCNC*, vol. 3, no. C, pp. 1609–1614, 2003.
- [50] S. David, "Approximation Algorithms for Combinatorial Problems *," J. Comput. Syst. Sci., vol. 278, pp. 256–278, 1974.
- [51] S. Rahili, J. Lu, W. Ren, and U. Al-Saggaf, "Distributed Coverage Control of Mobile Sensor Networks in Unknown Environment Using Game Theory: Algorithms and Experiments," *IEEE Trans. Mob. Comput.*, vol. 17, no. 6, pp. 1303–1313, 2017.
- [52] N. Bulusu, J. Heidemann, D. Estrin, and M. Rey, "GPS-less Low Cost Outdoor Localization For Very Small Devices," no. October, pp. 1–7, 2000.
- [53] Y. Qu and S. Member, "Relocation of Wireless Sensor Network," WAMICON 2011 Conf. Proc., pp. 1–5, 2011.
- [54] M. Abo-zahhad and S. M. Ahmed, "Coverage Maximization in Mobile Wireless Sensor Networks Utilizing Immune Node Deployment Algorithm," 2014 IEEE 27th Can. Conf. Electr. Comput. Eng., pp. 1–6, 2014.
- [55] C. Costanzo, V. Loscrí, E. Natalizio, and T. Razafindralambo, "Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network," *Comput. Commun.*, vol. 35, no. 9, pp. 1047–1055, 2012.
- [56] J. Guo and H. Jafarkhani, "Movement-efficient Sensor Deployment in Wireless Sensor Networks," 2017.
- [57] S. Ehsan and B. Hamdaoui, "A Survey on Energy-Ef fi cient Routing Techniques with QoS

Assurances for Wireless Multimedia Sensor Networks," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 2, pp. 265–278, 2012.

- [58] G. Wang, S. Member, and G. Cao, "Movement-Assisted Sensor Deployment," *IEEE Trans. Mob. Comput.*, vol. 5, no. 6, pp. 640–652, 2006.
- [59] M. Darienzo, M. Iacono, S. Marrone, and R. Nardone, "Estimation of the energy consumption of mobile sensors in wsn environmental monitoring applications," *Proc. - 27th Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2013*, pp. 1588–1593, 2013.