Less is More

DATA REDUCTION **in Wireless Sensor Networks**

Alireza Masoum

Less is More - Data Reduction in Wireless Sensor Networks

Alireza Masoum

Graduation committee:

Chairman:	Prof.dr. J.N. Kok
Promoter:	Prof.dr.ing. P.J.M Havinga
Assistant Promoter:	Dr.ir. N. Meratnia

Members:

Prof.dr.ir. R.N.J.Veldhuis	University of Twente
Dr.ir. M. de Graaf	University of Twente
Prof.dr. S. Baydere	Yeditepe University
Prof.dr. J. Bapat	International Institute of Information Technology Bangalore
Prof.dr. D. Das	International Institute of Information Technology Bangalore

This research is supported by (i) the Dutch Senter Novem Point One project "Free: True Wireless mesh networks for transport and logistics" (ii) the EU FP7-ICT project "GENESI: Green sEnsor NETworks for Structural Monitoring".

UNIVERSITY OF TWENTE.
DIGITAL SOCIETY INSTITUTE

DSI Ph.D. Thesis Series No. 18-001
Digital Society Institute
University of Twente
P.O. Box 217, NL – 7500 AE Enschede

ISSN 2589-7721

ISBN 978-90-365-4564-8

DOI 10.3990/1.9789036545648

<https://doi.org/10.3990/1.9789036545648>

Printed by Gildeprint, Enschede

Cover design: Mahdi Beheshti; Cover photo: Michael Wu

Copyright © 2018 Alireza Masoum, Enschede, The Netherlands

All rights reserved. No part of this book may be reproduced or transmitted, in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without the prior written permission of the author.

Less is More - Data Reduction in Wireless Sensor Networks

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
Prof. dr. T.T.M. Palstra,
on account of the decision of the graduation committee,
to be publicly defended
on Friday 8 June 2018 at 12:45 by

Alireza Masoum

born on the 19 September 1980
in Urmia, Iran

This dissertation has been approved by:

Prof.dr.ing. Paul J.M. Havinga (promoter)

Dr.ir. Nirvana Meratnia (assistant promoter)

Abstract

Wireless sensor networks are monitoring systems consisting of many small, low-cost and low-power devices called sensor nodes. A large number of sensor nodes are deployed in an environment to monitor a physical phenomenon, execute light processes on collected data, and send either raw data or processed information to the base station. Energy consumption is the main challenge of data collection in a wireless sensor network. Several energy efficient strategies are developed to ensure the longevity of a network.

Data reduction is one of the most significant energy management strategies in sensor networks. It concentrates on reducing the volume of the data collected, processed or communicated within the network. Proposed data reduction based energy management techniques assume the radio communication is the most significant energy consumption parameter. However, there are applications in which the computational and sampling energy costs are comparable to or even higher than the communication cost. Therefore, besides the communication level, there is a need to reduce energy consumption costs on sensing and computation levels as well.

The main focus of this thesis is to study quality aware data reduction techniques that improve data accuracy and energy efficiency in sensing and computation. Reducing the amount of data in these levels consequently reduces data transmission costs. Data reduction in sensing level is addressed by the adaptive sampling techniques which minimizes the number of sensing operations while data quality metrics are met. In the computation and communication levels, we use compressive sensing techniques to simplify data encoding in the sensor node level, efficiently compress data and reconstruct accurate data in the base station. These two objectives in turn reduces data transmission costs as well.

The main contributions of this thesis can be summarized as follows:

Adaptive sampling- state of the art: We present a comprehensive review of the state of the art in the field of adaptive sampling. We categorize existing solutions into centralized and decentralized methods and survey different adap-

tive sampling methods and their specific requirements. We compare the performance of existing adaptive sampling methods based on quality-of-service parameters such as energy efficiency, scalability, delay and data quality.

Spatial-temporal correlation based adaptive sampling: We introduce a spatial-temporal correlation-based adaptive sampling method which reduces demand for energy while meeting data accuracy requirements. We address energy efficiency by adjusting the number of active nodes and their sampling frequency.

Reward and punishment based adaptive sampling: Most of the existing adaptive sampling methods ignore cooperation among sensor nodes to determine a proper sampling frequency for the entire network. We address a reward and punishment-based cooperative adaptive sampling method to satisfy both energy efficiency and data-quality requirements. Sensor nodes are allowed to decide whether to change their sampling frequency based on network-wide energy level and global data behavior pattern changes. We encourage the sensor nodes to cooperate with each other to find the right balance between these two parameters by utilizing a reward and punishment system.

Data compression- state of the art: We present a comprehensive review of data compression techniques. We classified existing compression methods into lossless and lossy methods and then sub-divided each of those classes into local and distributed methods. We used energy efficiency, scalability, compression ratio and data-quality parameters to study and evaluate the performance of the solutions.

On analysis of transform basis in compressive sensing: We studied the effects of different transform bases on compressive sensing. Selecting an appropriate transform base for compressive sensing is crucial to sparse representation of signals, which can lead to fewer measurements and more accurate signal reconstruction. We present a number of transform bases and analyse their effect on compressive sensing performance in terms of signal-to-noise ratio (SNR) and signal reconstruction error.

Energy efficient distributed compressive sensing: We propose a distributed compressive sensing approach, which utilizes spatial correlation among sensor nodes to group nodes into coalitions. After the coalitions are formed, the proposed spatial-temporal correlation-based approach to compressive sensing is used inside each coalition to schedule sensor nodes and encode their readings. On the other hand, the base station employs a two-step joint sparsity-based recovery algorithm to reconstruct the original signal.

Samenvatting

Draadloze sensornetwerken zijn monitoringssystemen die bestaan uit vele kleine, goedkope en energiezuinige apparaten die sensornodes worden genoemd. Een groot aantal sensornodes wordt ingezet in een omgeving om een fysisch fenomeen te bewaken, simpele bewerkingen uit te voeren op de verzamelde gegevens, en ruwe data of verwerkte informatie naar het basisstation te verzenden. Energieverbruik is de grootste uitdaging voor het verzamelen van gegevens in een draadloos sensornetwerk. Verschillende energie-efficiënte strategieën zijn daarom ontworpen om de levensduur van een netwerk te garanderen.

Datareductie is een belangrijke strategie die gebruikt wordt voor het verminderen van het energiegebruik in sensornetwerken. Het concentreert zich op het verminderen van het volume van de gegevens die worden verzameld, verwerkt of gecommuniceerd binnen het netwerk. Veel gebruikte technieken voor gegevensreductie gaan ervan uit dat de radiocommunicatie de meest significante parameter is voor het energieverbruik. Er zijn echter diverse toepassingen waarbij het energieverbruik voor het bemonstering en verwerken vergelijkbaar zijn met, of zelfs hoger zijn dan de communicatiekosten. Daarom is het ook belangrijk om de energiekosten die gepaard gaan met de bemonstering en verwerking ook in ogenschouw te nemen, in samenspraak met de energiekosten voor de communicatie.

De focus van dit proefschrift is om kwaliteitsbewuste datareductietechnieken te bestuderen die de nauwkeurigheid van de gemeten gegevens en bijbehorende energie-efficiëntie in bemonstering en verwerking verbeteren. Het reduceren van de hoeveelheid gegevens in deze niveaus vermindert tevens de energiekosten voor de datatransmissie. Gegevensvermindering op het bemonsteringsniveau wordt geadresseerd door adaptieve bemonsteringstechnieken die het aantal waarneembewerkingen minimaliseren terwijl de kwaliteitseisen worden bereikt. In de berekenings- en communicatieniveaus gebruiken we compressive sensing technieken om codering op het niveau van sensornodes te vereenvoudigen, de data efficiënt te comprimeren en vervol-

gens deze data in het basisstation te reconstrueren. Deze twee doelstellingen verminderen op hun beurt ook de gegevensoverdrachtskosten.

De belangrijkste bijdragen van dit proefschrift kunnen als volgt worden samengevat:

Adaptieve bemonstering: we presenteren een uitgebreid overzicht van de stand van de techniek op het gebied van adaptieve bemonstering. We categoriseren bestaande oplossingen in gecentraliseerde en gedecentraliseerde methoden en onderzoeken verschillende adaptieve bemonsteringsmethoden en hun specifieke vereisten. We vergelijken de prestaties van bestaande adaptieve steekproefmethoden op basis van quality-of-service-parameters zoals energie-efficiëntie, schaalbaarheid, vertraging en datakwaliteit.

Ruimtelijk-temporele op correlatie gebaseerde adaptieve bemonstering: We introduceren een ruimtelijk-temporele op correlatie gebaseerde adaptieve bemonsteringsmethode die de vraag naar energie vermindert terwijl wordt voldaan aan de nauwkeurigheid van gegevensvereisten. We pakken energie-efficiëntie aan door het aantal actieve knooppunten en hun bemonsteringsfrequentie aan te passen.

Op beloning en straf gebaseerde adaptieve bemonstering: De meeste van de bestaande adaptieve bemonsteringsmethoden negeren de samenwerking tussen sensornodes om een juiste bemonsteringsfrequentie voor het gehele netwerk te bepalen. We onderzochten een op beloning en straf gebaseerde cooperatieve adaptieve steekproefmethode om te voldoen aan zowel energie-efficiëntie als datakwaliteitsvereisten. Sensornodes mogen beslissen of ze hun bemonsteringsfrequentie willen wijzigen op basis van het netwerk-brede energieniveau en veranderingen in de gegevens die ze verkrijgen van hun sensoren. We moedigen de sensorknopen aan om met elkaar samen te werken om de juiste balans tussen deze parameters te vinden door gebruik te maken van een belonings-en strafsysteem.

Datacompressie: we presenteren een uitgebreid overzicht van datacompressietechnieken. We classificeerden bestaande compressiemethoden in lossless en lossy-methoden en verdeelden elk van die klassen vervolgens in lokale en gedistribueerde methoden. We hebben energie-efficiëntie, schaalbaarheid, compressieverhouding en gegevenskwaliteitsparameters gebruikt om de prestaties van de oplossingen te bestuderen en te evalueren.

Over analyse van transformatiebasis in compressieve sensing: We onderzochten de effecten van verschillende transformatiebasissen op compressive sensing. Het selecteren van een geschikte transformatiebasis voor compressive sensing is cruciaal voor het weergeven van signalen, wat kan leiden tot minder metingen en een meer accurate signaalreconstructie. We presenteren

een aantal transformatiebasissen en analyseren hun effect op de prestaties van compressive sensing in termen van signaal-ruisverhouding (SNR) en signaal-reconstructiefouten.

Energie-efficiënte gedistribueerde compressive sensing: we stellen een gedistribueerde compressieve detectie-benadering voor, waarbij gebruik wordt gemaakt van ruimtelijke correlatie tussen sensornodes om die in coalities te groeperen. Nadat de coalities zijn gevormd, wordt de voorgestelde ruimtelijk-temporele op correlatie gebaseerde benadering van compressieve waarneming binnen elke coalitie gebruikt om activatie van sensornodes te plannen en hun bemonsteringen te coderen. Het basisstation gebruikt vervolgens een algoritme om het oorspronkelijke signaal te reconstrueren.

Acknowledgements

First and foremost, praises and thanks to the God, the Almighty, without His will I would have never found the right path. I thank Him for enlightening my soul with the respected love and compassion for the other humans and allowing me to enter a field where I could practice this desire. I ask sincerity in all my actions from Allah and I quote the verse from the Holy Quran “Say, My prayer, my offering, my life and my death are for Allah, the Lord of all the world” (Surat Al-’An’am, verse 162).

This thesis would not have been possible without the inspiration and support of a number of wonderful individuals — my thanks and appreciation to all of them for being part of this journey and making this thesis possible.

I would like to thank my supervisors Paul Havinga and Nirvana Meratnia for guiding me and, most importantly, for giving me the priceless freedom to choose my own path.

To all Pervasive Systems group members, I would like to leave here my gratitude for providing such a nice international working environment. I have learned uncountable new things during our PS discussion meetings, events, and lunch times.

I would like to thank Prof. Veldhuis, Dr. de Graaf, Prof. Baydere, Prof. Bapat and Prof. Das for accepting being part of my committee. I feel honored to have such experts in my defense.

I have to thank *Hamed* and *Meysam* for accepting to be my paranymphs. I am so grateful to *Mahdi* for designing the thesis cover.

I would like to thank many friends of mine around this globe who are in contact with me and are helpful all the time. Meysam (Etefaghan manam hamino migam), Hajar and lovely Dora (Aman Aman); Hamed (Bia avval pish farzamun ro yeki konim), Maryam and lovely Reihanah (Daghlar Gizi); Mehdi(Ina hamash azole ast), Hadis and lovely Taha; Majid(Bache meidune mojasame), Mohammad Hossein (Moaddabe topol) and Akram, Amir(Goodi), Mohammad Hekmat(asabem yokhde bakh): Thank you all for being there for me whenever I need it. Thank you for laugh- ing with me when God blesses

us with a funny moment, and crying with me when God is trying to tell me something. Thank you for being the one I share my fondest memories with: all the nights we stayed up really late, all the exciting adventures we went on and all the inside jokes we still laugh about today. Thank you for all the memories I've shared with you, and I can't wait for what crazy adventure we're going to go on next.

I am afraid that this sections is too short to name all my other friends, but I would list a couple of more friends: Reza Azarderakhsh, Rouzbeh, Jamal and Rezvaneh, Mohammadreza and Faeghe, Amin and Reihaneh, Bagher and Behnaz, Saeed and Vahan, Milad and Moloud, Hamid and Maryam, Jalal and Sara, Hamed and Malihe, Afshin and Elham, Mohammad and Neda, Alireza, Morteza and Mojhdeh, Pouria and Marzieh, Siavash and Mitra, Ali and Shokoufeh, Jalal, Fatemeh, Majid.

Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support. *Masoumeh* Nothing compares to the joy of having a sister like you. Thank you for always being there beside me through thick and thin. I would like to thank my brother and his family. My brother, *Reza*, Thanks for always being the perfect brother who is protective without being suffocative, liberal without being careless and watchful without being stifling. My dear *Arian*, I just want to reiterate how much I love you. I want you to grow up and be all that you wish for, and don't let anyone get in the way of it. I am grateful to my sister *Akram* for always being there for me as a friend. I was away; but you were there for Dad and Mom. Thank you for taking the huge responsibility of caring for them during my absence. Words will never be enough to show my appreciation to you.

My heart felt regard goes to my father in law and mother in law. To me, you are wonderful people I can go to for advice and support or simply a chat. I should tell you how grateful I am that you are both in my life. I also want to thank my sister in law *Reihaneh* and her family, brothers in law *Amin* and his family, *Ali* and his family for their love and moral support.

I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am. *Agha, Mama*, I feel so honored and blessed to have you as my parents, and want to express my gratitude for your care and support over the years. Thank you for instilling me with a strong passion for learning and for doing everything possible to put me on the path to greatness. I will never forget the important values you have passed down to me—particularly perseverance and honesty.

Agha (my daddy) It's been slightly more than 9 years since I last saw you, spoke to you, touched your hand, hugged you or just sat in your presence. I

miss you! I really, really miss you. I really wish I could hug you right now. I miss your words of wisdom, your sense of humour (I think you'd like my sense of humour, though I tend to be a tad cynical) and I miss the sound of your laughter. I do get to make duaah for you, though. That's something. It's the only thing I can do, really. I also get to hold on to the memories. I get to look at photos of you. I get to be proud when people still talk about what a fantastic man you were. You've left people with a lot of good memories of you. They always mention how you made them laugh, always listened, how you always helped whoever you could with whatever you could. Your generosity and selflessness is inspiring and unmatched.

Mama, you are the spunkiest, most loving, and kindest person I know. Everyone knows that I am your SON. But only I know that you are my SUN – lighting up my life with your hugs and smiles. You came from nothing and made yourself into something spectacular. Your drive and passion for life inspires me every day. Thank you for being you. You are beautiful inside and out, yet, it is your smile that radiates the sprouting waves of joy and happiness around every room that you enter.

Zahra, I cannot forget the day you say “yes” to me and through the years it has been a wonderful journey with you. I thank God every day that He heard my prayers and gave you to me as my wife. Thank you for being my voice of reason, my heart of the matter, and my sounding board. Thank you for always helping me think clearly, for helping me find the answers to my questions, and for giving me the courage to try. You have always believed in me and my capabilities, and you never doubted just what I can do. I know how blessed I am to be with someone as beautiful, intelligent, kind, and loving as you. Our lives are so happy and rich because of you. I am loved because of you. Thank you for everything that you do, which you do so tirelessly and lovingly. You make it look so easy and natural. There's truly no one else quite like you, my love. All my actions, decisions, battles, thoughts, plans and dreams lead to only one destination – you. Thanks Zahra.

Alireza Masoum
23-May-2018
00:02 AM

Contents

1	Introduction	1
1.1	Applications relevant to this thesis	2
1.2	Sensor network restrictions and challenges	5
1.3	Quality of service in wireless sensor networks	6
1.4	Energy efficiency in wireless sensor network	7
1.4.1	MAC layer optimization	7
1.4.2	Data routing	8
1.4.3	Data reduction	9
1.5	Research objectives	10
1.5.1	Adaptive sampling	12
1.5.2	Compressive sensing	12
1.6	Research question	13
1.6.1	Hypotheses	13
1.7	Contributions	14
1.8	Organization of the thesis	16
I	Adaptive Sampling	19
2	Adaptive Sampling: State of the Art	23
2.1	Introduction	23
2.2	Quality of service parameters for adaptive sampling	24
2.3	A taxonomy of adaptive sampling approaches	25
2.4	Centralized adaptive sampling	26
2.4.1	Temporal correlation-based centralized adaptive sampling	26

2.4.2	Spatial correlation-based centralized adaptive sampling	29
2.4.3	Spatial-temporal correlation-based centralized adaptive sampling	30
2.5	Decentralized adaptive sampling	32
2.5.1	Temporal correlation-based decentralized adaptive sampling	32
2.5.2	Spatial correlation-based decentralized adaptive sampling	35
2.5.3	Spatial-temporal correlation-based decentralized adaptive sampling	35
2.6	Hybrid adaptive sampling	37
2.7	Chapter summery	39
3	Spatial-Temporal Correlation-based Adaptive Sampling	41
3.1	Introduction	41
3.2	Preliminaries	42
3.2.1	Network architecture	42
3.2.2	Energy model	43
3.2.3	Data quality metric	44
3.2.4	Region definition	45
3.3	Overview of proposed approach	46
3.4	An energy-efficient adaptive sampling approach	48
3.4.1	Sub-clustering	49
3.4.2	Building the data prediction model	51
3.4.3	Sampler node selection	53
3.4.4	Temporal correlation-based sampling rate adjustment . .	57
3.5	MSSL adjustment	63
3.6	Performance evaluation	63
3.6.1	Evaluation of the temporal correlation-based adaptive sampling	64
3.6.2	Evaluation of spatial-temporal correlation-based adaptive sampling	71
3.7	Chapter summery	78
4	Reward and Punishment-based Cooperative Adaptive Sampling	81
4.1	Introduction	81

4.2	Preliminaries	82
4.2.1	Network architecture	82
4.2.2	Data behavioural states	82
4.3	Reward and punishment-based adaptive sampling	83
4.3.1	Initialization	83
4.3.2	Finding new sampling frequency	86
4.4	Performance evaluation	89
4.4.1	Prediction error	91
4.4.2	Number of samples	91
4.4.3	Effect of threshold	92
4.4.4	Trade-off between energy and error	94
4.5	Chapter summery	94

II Compressive Sensing

97

5	Data Compression: State of the Art	101
5.1	Introduction	101
5.2	Lossless data compression	102
5.2.1	Local lossless data compression	103
5.2.2	Distributed lossless data compression	105
5.3	Lossy data compression	108
5.3.1	Local lossy data compression	109
5.3.2	Distributed lossy data compression	110
5.3.3	Compressive sensing	114
5.4	Chapter summery	119
6	On Analysis of Transform Basis in Compressive Sensing	121
6.1	Introduction	121
6.2	Preliminaries	122
6.2.1	Sparsity	122
6.2.2	Compressive sensing	123
6.3	Sparse transform basis	125
6.3.1	Classification	125
6.3.2	Transform basis	125
6.4	Analysis of transform basis: A case study	129
6.4.1	Structural health monitoring	129
6.5	Impact analysis of transform basis on compressive sensing	131
6.6	Chapter summery	133

7	Energy Efficient Distributed Compressive Sensing	135
7.1	Introduction	135
7.2	Preliminaries	137
7.2.1	Distributed compressive sensing	137
7.2.2	Belief propagation	138
7.2.3	Network model and assumptions	141
7.3	Overview of the proposed approach	142
7.4	Coalition formation	144
7.4.1	Measurement matrix	146
7.4.2	Utility function	147
7.4.3	Coalition formation algorithm	150
7.5	Data gathering inside coalitions	151
7.5.1	Number of active sensor nodes	152
7.5.2	Compressive sensing based data gathering	153
7.5.3	Data gathering trees	155
7.6	A two steps signal recovery procedure	157
7.6.1	Sensor node signal model	159
7.6.2	Problem formulation	160
7.6.3	A belief propagation-based recovery algorithm	162
7.7	Performance evaluation	166
7.7.1	Assumption	167
7.7.2	Comparing distributed and individual compressive sensing	167
7.7.3	Comparing with other compressive sensing based methods	168
7.8	Chapter summery	172
8	Conclusion	175
8.1	Summary of contributions and lessons learned	175
8.2	Future research directions	179
	Bibliography	181
	About the author	201

CHAPTER 1

Introduction

Wireless sensor networks are revolutionary new monitoring platforms consisting of many small, low-cost and low-power devices called sensor nodes. A large number of sensor nodes are deployed in an environment to monitor a physical phenomenon, execute light processes on collected data, if needed, and send either raw data or processed information to the base station. The base station sometimes acts as a gateway and relays the collected data to a remote location. In other cases, the base station is the destination node, which gathers, processes and stores collected data [1, 2].

From a physical architecture point of view, sensor nodes essentially consist of processing, power, memory, radio and sensing units (Figure 1.1). The sensing unit deals with physical criteria, while the processing unit analyses sampled data, and the memory unit stores data. Sensor nodes use the radio units to communicate with other nodes through a wireless network infrastructure. However, all these units are powered by a limited power unit. Since sensor nodes are resource-restricted, it is essential that they have light-weight and simple functions [2].

Wireless sensor networks employ single-hop or multi-hop communication to transfer data. In single-hop communication, each sensor node has direct access to either the base station or the cluster head and sends its data directly to them. However, there are applications in which sensor nodes do not have direct access to the base station. In these cases, multi-hop communication is used, whereby each sensor node acts as a relay node and forwards the data received by its neighbor nodes to the destination node [3].

From the deployment perspective, sensor node distribution is often ad hoc and unstructured. After being deployed in the environment, sensor nodes often may not be accessed, and thus can be neither maintained nor controlled (e.g. replacing batteries). Therefore, energy consumption is one of the most im-

Figure 1.1: Sensor node architecture

portant challenges in these networks. The issue of energy has understandably received considerable critical attention in studies of wireless sensor networks.

1.1 Applications relevent to this thesis

Wireless sensor nodes can be equipped with sensors for various specifications such as temperature, humidity, vibration, light, and so forth. Thanks to the diversity of sensors, these networks can support different types of data-monitoring and tracking applications. The applications which are of particular relevance to this thesis are described below:

- **Environmental monitoring** is an important application, which may have an important impact in our daily life. For example, climate change as a critical global phenomenon illustrates the necessity of environmental temperature monitoring. The climate change lead to very serious environmental effects like breaking of sea ice, increasing in sea water level, heat waves, glasier melting. Another reason for global warming is the Greenhouse effects which heats the surface of earth. Therefore a monitoring application is needed to guarantee the stabilization of the envi-

Figure 1.2: Enviromental monitoring applications

ronment. Indoor monitoring applications measure parameters like temperature, humidity, light, air-quality and CO₂ inside private and public buildings while outdoor monitoring measures those parameters in the nature. Smart houses and offices equipped with automated air conditioning/heating system and connected devices provide comfortable and convenient working and living environments for humans. Warehouse, logistics, cold chain and data centers air condition monitoring and controlling are other examples of industrial applications. Habitat monitoring, agriculture monitoring and weather forecasting application are the examples of outdoor monitoring applications. Among all environmental parameters, temperature is one of the most simple yet critical one in many of these applications. For example, in cold chain applications, controlling the environmental data is a key parameter to keep the freshness of foods and fruits inside warehouse and during transportation. In case of data centers, it is essential to keep the indoor temperature in a certain range in which servers can continue working with the highest perfor-

Figure 1.3: Structural health monitoring applications

mance. Investigation and analyzing of global warming phenomenon requires precise analysis of environmental temperature data. The solutions presented in this thesis target these applications.

- **Structural health monitoring** is an emerging monitoring application focused on the field of civil structures like bridges, tunnels, aerospace ships, dams, pillars and roads to continuously monitor the condition and longevity of these structures. Due to the events like aging of these structures, earthquake, lack of proper maintenance and construction error, it is more likely that these structures face damages which can lead to collapse. The I-35W Mississippi River bridge collapse in 2007 [4] is one of the structure collapse examples which killed 13 people and injured 145. Penmanshiel tunnel collapse near Grantshouse, Berwickshire in 1979 [5] and Rastatt tunnel collapse in 2017 [6] are two other examples of structural disasters. To manage and avoid these disasters, structural health monitoring is suggested to continuously monitor these structures, early

detect problems, detect and locate damages which lead to predict and avoid collapse, improve public safety, reduce maintenance and operation costs. This monitoring is accomplished by populating a given structure with a large amount of sensor nodes to cover the complexity and large scale of this structures. These sensor nodes extract and produce a massive amount of data streams which need efficient data reduction technique to reduce the data gathering and communications costs. The second part of this thesis target this applications as well as environmental monitoring applications.

1.2 Sensor network restrictions and challenges

Wireless sensor networks carry certain challenges and restrictions in terms of their resource limitations, data redundancy, network dynamics, energy balancing, environmental conditions, application requirement diversity and scalability. These constraints are described briefly below [3, 7]:

- **Resource limitation:** These include, among other things, energy, bandwidth, memory, buffer size, processing and transmission power. Energy is the main restriction in sensor networks and is addressed in the majority of sensor network studies.
- **Data redundancy:** The spatial and temporal redundancy inherent to wireless sensor networks produces a large amount of redundant data, which can be employed to improve data reliability; however, sensing, processing and transmitting this data leads to higher energy costs.
- **Network dynamics:** Node failure, link failure, node mobility, and so forth, can change the network topology. Unpredictable conditions in the environment where sensor nodes are deployed can also result in frequent changes of the network topology.
- **Energy balancing:** Wireless sensor nodes come in various types and operate over various periods of activity. This means that some nodes must remain active for long periods, while others have low activity and spend most of their time in sleep mode. Being in active mode requires energy, and the longer the active period, the more energy is spent. This may result in the exhaustion of some of the active nodes and create holes in the network. An effective energy-balancing and distribution mechanism or energy-harvesting technique can greatly prolong network lifetime.

- **Application requirement diversity:** Sensor networks support different types of applications, which produces elastic (periodic) and inelastic (real-time) traffic. For example, event-based applications produce burst traffic, while monitoring and target-tracking applications produce periodic and data-stream traffic, respectively. Therefore, sensor networks must be able to support different types of traffic.
- **Environmental conditions:** Sensor nodes can be deployed in different environments, including underwater, in the human body, in urban areas, and so forth. The diversity of the environmental conditions requires different levels of sensor-node capabilities.
- **Scalability:** Wireless sensor network size is likely to grow with increasing number of sensor nodes and network load. The number of sensor nodes joining or leaving network can be variant which may create lots of scaling issue in the network. Therefore, proposed protocols, algorithms and strategies should respect scalability concern of these networks.

1.3 Quality of service in wireless sensor networks

Compared to applications in wireless ad hoc networks, wireless sensor network applications have additional requirements regarding quality of service. Providing acceptable quality of service over a wireless sensor network requires taking into account not only the resource constraints of individual nodes and the entire network, but also physical node failure, lack of communication reliability, and the need to assess dynamic network performance [8].

In typical warehouse applications with time-sensitive goods; for instance, sensor nodes must quickly and reliably report conditions such as humidity and temperature. When unwanted conditions such as spoilage or extreme temperature variations arise, sensor nodes must immediately inform the base station and possibly generate an alarm. If data is delayed on the way to the base station, then the situation may become uncontrollable. Delays in data transmission from sensor nodes to the base station and packet loss during transmission may cause system performance to deteriorate. Such delays and information loss cannot be tolerated in many of the safety-critical deployments of wireless sensor networks [9, 10].

Although the quality of service requirements are application-dependent, many applications are concerned with reliability, real-time actions, robustness,

trustworthiness and adaptability. Various quality of service metrics may be used to evaluate the degree to which these requirements are met.

Quality of service in wireless sensor networks can be studied with reference to the application or the network level[8]. Application-level quality of service consists of the application's requirements and expectations of the network, measured based on the services that the network is to deliver. The application level is not concerned about how the network provides this service. Network-level quality of service refers to how network resources are managed to satisfy the application.

Various quality-of-service criteria have been defined for sensor networks over these two levels, i.e., priority, latency, lifetime, connectivity, coverage, survivability, fault tolerance, reliability, confidentiality, safety, availability, maintainability, energy efficiency, throughput, adaptability, scalability, fault recovery, jitter, packet loss and error rate to name a few.

1.4 Energy efficiency in wireless sensor network

As previously stated, energy efficiency is a continuing concern in wireless sensor networks. To ensure the longevity of a network, energy-efficient techniques are essential. Proposed energy-efficient solutions for sensor networks can be divided into media access control (MAC) layer optimization, data routing and data reduction solutions [9]. Each of these methods is described briefly below.

1.4.1 MAC layer optimization

The MAC layer controls and manages channel access, scheduling and queuing policies. Given that radio transmission demands more power than other tasks, designing and implementing energy-efficient MAC protocols is the optimal solution to prolong network lifetime [11, 12, 13]. MAC protocols may be unscheduled or scheduled.

1.4.1.1 Unscheduled MAC Protocols

Unscheduled or contention-based MAC protocols do not consider any scheduling or reservations [14]. Therefore, sensor nodes contend for access to the shared bandwidth in order to send their data. This category offers lower memory use, power consumption and message overheads, which leads to more

scalable and adaptable protocols. However, contention-based access also faces collision, idle listening and overhearing issues, which cause a higher degree of latency, energy expenditures and greater packet loss [15, 13]. Carrier-sense multiple access (CSMA) exploits collision avoidance mechanisms and a carrier sense mechanism in order to mitigate these effects [14]]. The main categories of unscheduled MAC protocols are multi-channel [16, 17, 18], event-driven [19, 20] and preamble based MAC protocols [21, 22, 23].

1.4.1.2 Schedule-based MAC protocols

Organizing orderly communication among sensor nodes is the main concern of protocols. For scheduling, one central point broadcasts a scheduling message in order to inform each sensor node about the channel access time. Because they do not engage in contention, these protocols are adaptable and scalable to different traffic types and network scenarios. Furthermore, dedicated and proper scheduling ensures bounded latency [13]. Unnecessary idle listening and collision are avoided, which makes these protocols more energy efficient [13, 12, 24]. Slotted contention [25, 26, 27], time-division multiple access (TDMA) [28, 29, 30] and reservation-based protocols [31, 32, 33] are the main types of scheduled-based MAC protocols.

1.4.2 Data routing

Finding and establishing routes in order to transfer data from sensor nodes to the base station is the prime role of data-routing policies. According to the application requirements, routing mechanisms must provide an energy-efficient, time-sensitive and reliable data delivery. The unique sensor constraints outlined previously necessitate different types of routing algorithms for sensor networks compared with ad hoc and wireless networks [34]. Energy-efficient routing solutions include flat, hierarchical and location-based routings. The flat approach assigns the same role or functionality for all nodes, while hierarchical and cluster-based approaches consider different roles for different nodes [35, 36].

1.4.2.1 Flat routing protocols

In flat based routing protocols, sensor nodes play the same role and cooperate with each other to collect data. Sensor protocols for information via negotiation (SPIN) [37], directed diffusion [38], rumor routing [39], minimum

cost forwarding algorithm (MCFA) [40], information-driven sensor querying (IDSQ) [41], constrained anisotropic diffusion routing (CADR) [41], COUGAR [42], ACQUIRE [43] and random walks [44] are the main types of flat based routing protocols.

1.4.2.2 Location based routing protocols

In location-based routing methods, each node must be aware of its location. Location information is essential for this algorithm to perform route discovery, maintenance and data-transmission tasks [45, 46]. Major types include restricted flooding based [45, 47], distance based [35, 48], probability based flooding [49, 50] and virtual portioning based [51, 52, 53] routing protocols.

1.4.2.3 Hierarchical routing protocols

Hierarchical routing protocols partition the network into clusters, each of which has one cluster head. Grouping sensor nodes into clusters has advantages such as localized routing, which improves energy efficiency and bandwidth utilization [54, 55]. Centralized single-hop [56, 57], centralized multi-hop [58, 46], distributed single-hop [59, 60, 61] and distributed multi-hop [62, 63] are the main types of clustering technique for hierarchical routing.

1.4.3 Data reduction

Data reduction is one of the most significant energy management strategies in sensor networks. It concentrates on reducing the volume of the data collected, processed or communicated within the network. Spatial-temporal correlation among sensor node readings leads to redundancy in both data-gathering and data-transmission procedures. Several studies have investigated data reduction solutions in order to decrease data sampling, processing and transmission costs for wireless sensor networks [64]. These solutions can be classified into three major categories: in-network data processing, data prediction and data compression.

1.4.3.1 In-network data processing

In-network data processing mainly focusses on data aggregation solutions combining data from different sources or nodes into a single entity [65, 66].

Data aggregation methods use aggregation functions such as SUM, AVERAGE and MIN/MAX to aggregate data and transfer only the aggregated results. This technique thus manages data reduction among the nodes while transferring data from sensor nodes to the base station [67, 68]. Data aggregation is often closely integrated with routing protocols; hence, most data aggregation studies aim to design data routing and forwarding solutions which expedite the aggregation process. Cluster-based [69, 70, 71], chain-based [72], tree-based [73, 74, 75] and multipath-based [76] data gathering are the major data aggregation types among these solutions.

1.4.3.2 Data prediction and acquisition

Data prediction and acquisition emphasizes building data estimation models and minimizing the number of samples in sensor nodes and the base station. Employing these prediction models, sensor nodes reduce their sampling frequency by skipping the observations that can be predicted at the base station, and this reduction lessens energy consumption. These prediction models are changed and updated dynamically so as to track and predict the environmental data change [77, 78, 79]. Therefore, these techniques are further divided into stochastic [80, 81], time series prediction [82, 83, 84] and adaptive sampling methods [85, 86, 87].

1.4.3.3 Data compression

Data compression is a process whereby the amount of data or transmission time is reduced [88]. Data compression algorithms aim to find an efficient way to compress data to reduce the energy costs of the individual sensor nodes and improve the overall ability of the whole network. However, accuracy must be guaranteed when data is decoded [89, 88]. Existing data compression techniques can broadly be classified into two categories; (i) lossless compression and (ii) lossy compression. Lossless data compression techniques, as the name indicates, compress data without any loss [90, 91, 92]. A lossy data compression technique is a compression technique in which some data is discarded during the compression process [93, 94, 95, 96].

1.5 Research objectives

Different applications employ different types of sensor node devices equipped with variety of sensing, computation and communication units. Due to variety

Table 1.1: Different sensor types and their energy consumptions

Sensor Type	Sensors	Imote2(<i>E_{sens}</i>)	Imote2(<i>E_{comp}</i>)	Mica2(<i>E_{sens}</i>)	Mica2(<i>E_{comp}</i>)	TelosB(<i>E_{sens}</i>)	TelosB(<i>E_{comp}</i>)
Acceleration	MMA-7260Q[100]	0.0000268	0.044	0.000017	0.096	0.0000268	4.01
Pressure	220/2600 Series[101]	0.00013	0.044	0.000079	0.096	0.00013	4.01
Light	ISL29002[102]	0.00068	0.047	0.00044	0.0106	0.00068	4.13
Proximity	CP18[103]	0.0267	0.047	0.17	0.105	0.267	4.12
Humidity	SHT1X(H)[104]	0.4	0.043	0.255	0.77	0.4	12.8
Temperature	SHT1X(T)[104]	1.5	0.94	0.957	2.65	1.5	37
Level	LUC-M10[105]	9.22	0.104	5.89	0.266	9.22	6.2
Gas	MICS-5521[106]	26.98	1.84	17.242k	5.2	26.98	69.9
Flow Control	FCS-GL1/2A4-AP8X-H1141[107]	97.2	0.46	62.1	1.28	97.2	19.4
CO2	GE/Telaire 6004[108]	1294.25	9.03	798.2	25.64	1249.25	333.8

of those units, the energy consumption of sensor nodes varies greatly. In case of sensing units, there are two types of sensors: passive sensors like pressure and light sensors with low power consumption and active sensors like CO2 and sonar sensors with significant power consumption. Authors in [62, 95] address a comparison among sampling, computation and communication energy costs for Imote2 [97], Mica2 [98] and TelosB [99] nodes which are equipped with different sensor types. This comparison is represented in Table 1.1. Basically, the comparisons are normalized based on the communication energy costs. According to this table, in some cases sensing and computation energy costs are equal or greater than communication cost. Proposed data reduction based energy management techniques assume the radio communication is the most significant energy consumption parameter. However, there are applications wherein the computational and sampling energy costs are comparable to the communication cost.

Data compression is a well known techniques in sensor networks to reduce data sampling, computation and transmission costs. Compression in sensor networks can be applied in sensing, data and communication levels: (i) compression in the sensing level reduces the number of sampling operation, (ii) compression in data level utilizes efficient encoding methods to reduce data size and (iii) compression in communication level minimizes data packet size. Most of the proposed solutions concentrate on compression in the communication level while assume sensing and computation costs are insignificant. On the other hand, these solutions employ excessively computationally complex algorithms for data encoding which are not suitable for the resource restricted sensor node characteristics.

The main focus of this thesis is to study the quality aware data reduction solutions that improve data accuracy and energy efficiency in sensing and computation levels. Reducing data in these levels consequently reduces data transmission costs. Data reduction in sensing level is addressed by the adaptive

sampling technique which minimizes the number of sensing operations while data quality metrics are met. In the computation and communication levels, we use compressive sensing technique to simplify data encoding in the sensor node level, efficiently compress data and reconstruct accurate data in the base station. These two objectives in turn reduces data transmission costs as well.

1.5.1 Adaptive sampling

The first part of this thesis concentrates on the compression at the sensing level and studies existing adaptive sampling solutions. During sensing procedure, sensor nodes may generate redundant samples which can waste communication, processing and memory resources. Adaptive sampling is an energy-efficient data-gathering method which determines how often a sensor node should sample during a specified time period (frequency) and which sensor node should perform the sampling (schedule) while satisfies data quality requirements. In this way, it removes the redundant samples by tuning the sampling frequency of the sensor nodes. Adaptive sampling methods find the correlation among data in the sensing field in order to adjust sensor node sampling frequency accordingly. Thanks to data correlation, adaptive sampling may ask only some sensor nodes to send data without degradation of accuracy. Furthermore, this form of sampling decreases communication and bandwidth costs [86, 109].

1.5.2 Compressive sensing

The second part of this thesis targets data compression at computation level and utilizes compressive sensing technique to minimize the data computation and consequently communication costs while satisfies data quality in term of data accuracy. Compressive sensing is a concept from the field of signal processing which is able to reconstruct sparse or compressible signals from a small number of measurements without having prior knowledge of the signal structure. Unlike classical compression techniques, compressive sensing simplifies the encoding procedure, while the decoding procedure remains complicated. This characteristic best suits wireless sensor network restrictions by employing simple encoding on the resource-restricted sensor nodes and a complex decoding procedure at the powerful base station [110, 111, 95]. Compressive sensing utilizes information rate instead of sampling rate to sample and recover the signal [95].

1.6 Research question

In pursuit of the research objectives, the following main research question will be answered in this thesis:

How can quality-aware data reduction be achieved in wireless sensor networks while fulfilling user-defined quality of service requirements in terms of data quality?

Considering various energy-efficient solutions, we approach our main research question by answering the following two sub-questions:

- **RQ.1:** *How often should a sensor node take samples during the specified time period (frequency), and which sensor nodes should do sampling (schedule) in order to meet quality-of-service requirements?*
- **RQ.2:** *How should a sensor node compress and reduce data size without sacrificing data quality? How should a base station reconstruct encoded data with minimal quality reduction?*

1.6.1 Hypotheses

To answer the first research question (**RQ.1**), we start from the hypothesis that in the high density sensor networks, temporal correlation in sensor node readings can be used to adjust sampling frequency in response to environmental data changes. Furthermore, the spatial correlation among sensor nodes can be exploited in order to achieve efficient selection of active sensor nodes to sample and forward data. To evaluate our solution, we need a high density network which represents high degree of correlation among sensor node readings. Furthermore, the observed data should contain both low and high variability levels. Dataset collected from the sensor nodes deployed in the Intel research laboratory at Berkeley [112], is a one of the most popular public dataset widely used in sensor networks area and has those required characteristics.

We address the second research question (**RQ.2**) with the hypothesis that most signals of interest in wireless sensor network applications have sparse representation. In sparse signals, a small number of measurements can represent the whole signal. With this in mind, we apply compressive sensing as an energy-efficient compression solution to sample and compress sensor node readings simultaneously. Furthermore, we distribute sensor node compression inside different coalitions, which improves scalability and energy efficiency parameters. We also propose a two-step data reconstruction solution in the base

station to recover the original data with minimal data-quality reduction. To evaluate this research question, we need a dataset which shows higher level of sparsity. We use the sensor scope dataset [113] which is well known in the field of compressive sensing due to its potential sparsity characteristic.

1.7 Contributions

This thesis proposes new data reduction techniques to improve sensor network quality of service specifications in terms of energy efficiency and data accuracy. The proposed solutions target two main data-reduction strategies: adaptive sampling and data compressive sensing. Therefore, this thesis is divided into two parts.

The first part, concerning adaptive sampling techniques, consists of Chapters 2-4.

- **Chapter 2** presents a comprehensive review of adaptive sampling techniques for wireless sensor networks. This study categorizes existing solutions into centralized and decentralized methods and surveys different adaptive sampling methods and their specific requirements. For the centralized and decentralized methods, we categorize the existing methods into temporal, spatial and spatial-temporal correlation-based approaches. We compare the performance of existing adaptive sampling methods based on quality-of-service parameters such as energy efficiency, scalability, delay and data quality.
- **Chapter 3** states that environmental data, for example temperature, usually changes gradually. It often follows a normal distribution and falls into a specific range. As long as data falls within such a given range, its behavior is assumed to be normal. Existing adaptive sampling techniques, however, often ignore this property and aim to identify any small changes in the data to meet the data-quality requirements of the application. As such, they deplete power in transmitting irrelevant data. In this chapter, we propose an adaptive sampling approach that leverages the benefit of this concept to achieve a trade-off between energy efficiency and data quality. While changes in sensor readings that fall within a given data range are not reported, those closer to the boundaries of the data range are expected to have more values. In other words, those changes may indicate that the future readings are quite likely to

fall outside the range and as such need to be transmitted. Having this in mind, we introduce a spatial-temporal correlation-based adaptive sampling method which diminishes demand for energy while meeting data accuracy requirements. We address energy efficiency by adjusting the number of active nodes and their sampling frequency. This approach is implemented in three phases: In the first phase, a K-means sub-clustering algorithm is utilized to group sensor nodes into the sub-clusters. In the second phase, an auto-regressive moving average (ARMA)-based data prediction model is introduced. Finally, each sensor node employs a dynamic sampling frequency adaption method to synchronize it with the data changes.

- **Chapter 4** states that most of the existing adaptive sampling methods ignore cooperation among sensor nodes to determine a proper sampling frequency for the entire network. Instead, each node decides its own sampling frequency based on its limited observation, or else the base station decides the sampling frequency for the sensor nodes. While the latter approach may lead to higher demand for energy due to higher data transmission, the former may lead to a global data-quality loss in order to lower demand. To address these two aspects simultaneously, this chapter addresses a reward and punishment-based cooperative adaptive sampling method to satisfy both energy efficiency and data-quality requirements. When a sensor node detects frequent environmental changes, it increases its sampling rate to adapt to those changes. Accordingly, other sensor nodes that do not experience the same changes decrease their sampling rates to balance energy consumption over the network. To motivate sensor nodes to cooperate with each other, a reward and punishment mechanism is suggested.

The second part of this thesis, focussing on compressive sensing techniques, includes Chapters 5-7.

- **Chapter 5** presents a comprehensive review of data compression techniques proposed for wireless sensor networks. This comprehensive study divides existing solutions into lossless and lossy data compression methods, and sub-categorizes these into local and distributed solutions. It then surveys various compression methods and their specific requirements and analyses their performance based on quality-of-service parameters such as energy efficiency, scalability, compression ratio and data quality.

- **Chapter 6** presents the effect of transform bases on compressive sensing performance in terms of signal-to-noise ratio (SNR) and signal reconstruction error. Selecting an appropriate transform base for compressive sensing is crucial to sparse representation of signals, which can lead to fewer measurements and more accurate signal reconstruction. To the best of our knowledge, the selection of an appropriate transform basis has not been adequately studied, as most existing compressive sensing techniques have focussed on optimizing reconstruction algorithms. This chapter classifies different signals based on their characteristics and theorises the assignment of an appropriate transform basis for each specific type of signal.
- **Chapter 7** proposes an energy-efficient approach to compressive sensing-based data-gathering. We introduce a distributed compressive sensing approach, which utilizes spatial correlation among sensor nodes to group nodes into coalitions. The coalition-formation method is represented by a block diagonal measurement matrix, with each diagonal entity corresponding to one of the coalitions. After the coalitions are formed, a proposed spatial-temporal correlation-based approach to compressive sensing is used inside each coalition to schedule sensor nodes and encode their readings. on the other hand, the base station employs a two-step joint sparsity-based recovery algorithm to reconstruct the original signal.

1.8 Organization of the thesis

The remainder of this thesis is organized as shown in Figure 1.4.

As we have seen, the first part concentrates on adaptive sampling solutions to answer the first research question (*RQ.1*). Chapter 2 gives an overview of the state-of-the-art adaptive sampling solutions by describing their characteristics. Chapter 3 describes our contribution for energy-efficient adaptive sampling-based data collection, and chapter 4 presents a cooperative adaptive sampling solution.

In the second part of this thesis, we present a compressive sensing solution to answer the research question (*RQ.2*). Chapter 5 presents the state of the art in compression techniques. Chapter 6 analyses transform base selection and its impact on compressive sensing. Chapter 7 describes an energy-efficient compressive sensing based data gathering method. Finally, Chapter 8 concludes this thesis with a summary and suggestions for future work.

Figure 1.4: Organization of the thesis

Part I

Adaptive Sampling

Figure 1.5: Organization of the thesis

Adaptive Sampling: State of the Art

2.1 Introduction

Monitoring of environmental phenomena in space and time is feasible through sampling approaches in wireless sensor networks. Sampling has a critical impact on the network lifetime: In general, the higher the sampling frequency, the higher the demand for energy, and consequently the shorter the network lifetime. Adaptive sampling is a strategy to sample data only when needed instead of at fixed intervals. Thus it can prolong the network lifetime. While adaptive sampling can improve the network lifetime, it may decrease data quality and detection accuracy for critical events. Therefore, it is important to find a good balance between energy consumption, data quality and detection accuracy in designing an adaptive sampling approach [1].

In high-density wireless sensor networks, neighboring nodes sense similar phenomena. Moreover, a node often senses the same phenomena over a period of time since environmental conditions change gradually. Therefore, adaptive sampling approaches adjust their sampling frequency based on the correlation (in space, time or both) between the data of sensor nodes. By doing so, they reduce the need for energy in sensing and transmission, as well as the required bandwidth [114].

Figure 2.1 illustrates the adaptive sampling procedure. In order to gain insight into the data pattern to tune the sampling rate, adaptive sampling approaches often use data modeling and prediction concepts.

Adaptive sampling approaches must satisfy a number of quality of service parameters that are application-dependent. The same quality of service parameters are used to evaluate each of them. In this section, we first explain the quality of service parameters used to compare existing adaptive sampling approaches.

Figure 2.1: An overview of the adaptive sampling procedure

2.2 Quality of service parameters for adaptive sampling

Almost none of the existing adaptive sampling approaches considers all quality-of-service parameters. Instead, they focus mainly on one or a few, despite the fact that evaluating the applicability of an adaptive sampling method requires one to take all parameters into account. The quality of service parameters relevant for adaptive sampling are:

- **Data quality:** representing the completeness and representativeness of the collected data. Data quality has two components, i.e, temporal consistency and numerical consistency. The former relates to the temporal gaps between collected data, while the latter relates to its representativeness. It also indicates the importance of missing values, i.e. whether an important data point indicating abnormal behavior has been missed. Numerical consistency is often referred to as data accuracy.
- **Energy efficiency:** representing the energy required to perform adaptive

Figure 2.2: Categories of adaptive sampling approaches

sampling. There is a direct relationship and thus trade-off between data quality and energy efficiency.

- **Computational complexity:** representing the resources required inside and outside the network to perform adaptive sampling. Running complicated data modeling and prediction methods on the sensor nodes may introduce a large overhead on the nodes or prove to be impossible because of their limited resources. Running them off the network also requires data transmission, which in turn comes at the cost of energy efficiency. Computational complexity has a direct relationship with energy efficiency and data quality.
- **Scalability:** how well an adaptive sampling approach performs as the network size increases. Proposed solutions should be scalable to the changes in the network size or workload.
- **Delay:** how fast adaptive sampling approaches report critical data indicating abnormal situations. There are some delay-sensitive applications like critical situation monitoring and security surveillance which can not tolerate timely data processing and transmission of information over the network.

2.3 A taxonomy of adaptive sampling approaches

In this section, we provide a taxonomy of existing adaptive sampling approaches and compare them in terms of how well they meet the quality of service requirements. Various criteria can be used to categorize adaptive sampling

approaches. Our main classification criterion is what will decide to change the sampling rate: the sensor nodes themselves or the base station. Given this criterion, we categorize adaptive sampling approaches into centralized and decentralized approaches. There are also hybrid approaches that make use of both. Under each of these categories, the existing approaches can be further classified into those that utilize spatial correlation, temporal correlation and spatial-temporal correlation in sensor data. This classification system is illustrated in Figure 2.2. The existing approaches for each class are reviewed below.

2.4 Centralized adaptive sampling

In centralized approaches, modeling, predictions, and computations required to find the sampling rate and the correct node for sampling are performed at the base station. The base station creates an initial prediction model and sends it to the sensor nodes. The sensor nodes sample data and compare it with their prediction model. If the error between their sampled data and the predicted value is high (greater than an acceptable threshold), they send their sampled data to the base station. The base station updates the prediction model and decides the sampling rate for the nodes. Since centralized approaches obtain global knowledge of the monitored phenomena and have more resources, they are able to provide accurate models. However, they are also prone to single point-of-failure risk. In other words, if communication between the sensor nodes and the base station fails, the prediction models can no longer be updated, and the outdated models will produce low data quality. These approaches also suffer from extra communication overhead (and consequently higher energy consumption) between the sensor nodes and the base station to keep the prediction models up to date. This increases in severity when both the network size and the dynamicity of the network increase. Temporal, spatial and temporal-spatial sampling approaches are examined individually below.

2.4.1 Temporal correlation-based centralized adaptive sampling

There is a degree of similarity between the consecutive readings of individual sensor nodes. This similarity is known as temporal correlation. Based on the temporal correlation among sensor readings, the sampling rate can be adjusted to provide a certain level of data quality. During each time period, the sensor node gathers data and sends it to the base station.

Previous studies [115, 86] address temporal correlation-based adaptive sampling in snow monitoring applications. At the beginning, the algorithm uses the initial observations to predict the maximum sampling rate and defines a threshold to detect changes in this rate. If the maximum sampling rate for the recent observation exceeds a predefined threshold, a new maximum rate is set. The base station estimates the sampling rates for the sensor nodes. They use the CUSUM test [116], which is a typical version of change detection test, i.e. a statistical measure of a stationary hypothesis for a process. The proposed mechanism modifies the CUSUM test in order to include non-stationary change detection to find the maximum frequency. Simulation results show that the method achieves good data quality, to the extent that the reconstructed data in the base station is very close to the real data. This method also decreases the number of data transmissions, which leads to low energy consumption.

Sensor nodes deliver continuous, high-volume, possibly noisy and time-varying data streams to the base station. Data-stream characteristics are important factors in changing the sampling rate. In this regard, network bandwidth can be assigned to the sensor nodes based on the extent of their activities. Several studies have utilized this concept to achieve better performance. One previous study [117] uses this concept to adapt the sampling rate and employs a Kalman filter for data prediction. This method is not scalable, and its high computational complexity comes at the expense of greater energy consumption to provide the high data accuracy.

In the approach reported in [118], the authors use backtracking models [119] in combination with an energy-aware routing approach to gather and reconstruct data. They use centralized flood predictor models with grid computing for the base station. This flood predictor model employs the ensemble Kalman filter [120] and a stochastic one-dimensional numerical hydraulic model to implement the adaptive sampling mechanism. The prediction model uses a data importance parameter to detect the sampling and transmission rates. The main parameters for the adaptive routing are the residual energy of the communication nodes, data priority and link cost. This technique suffers from a loose correlation between the adaptive sampling and the routing mechanisms. This looseness leads to the problem that if the intermediate nodes have a high sampling rate, their battery power is depleted rapidly, and consequently the data cannot be further relayed and will be lost. The prediction model has a high computation cost over a short time, but provides highly accurate data.

The authors of [121] propose a typical routing layer scheduling protocol in order to support adaptive sampling and provide different transmission rates.

Every node has a scheduling scheme based on its sampling rate. To support this scheme, a route-partitioning mechanism is developed, in which sensor nodes send route discovery packets to the base station. Every node placed along the path, in which discovery packets move towards the base station, appends its sampling rate and address to those packets. After receiving all route discovery packets, the base station establishes and manages different routes based on the current sampling rates. Furthermore, it finds the overlaps of different sampling rates in order to adjust them by considering the longest duty cycle. It schedules the activity time of each node by taking into consideration the packet generation time. This method supports various sampling rates, minimizes packet loss and reduces the delay. It also cuts power consumption by using sleep and wake states.

In [122], a partitioning algorithm is used to divide the sampling area into non-overlapping regions. Each sampling region then performs sampling and sends its readings to the base station, which combines the readings from different regions and constructs the final result. By comparing the sampling statistics with the final result in each region, the base station finds an optimal sampling rate for each region. In this way, different regions have different sampling rates and sampling costs, and this diversity can help achieve optimal accuracy and an acceptable level of energy consumption.

Another work [109] studies an adaptive sampling method for air-monitoring applications. They show that there are certain patterns and similarities in the daily air measurements which can be utilized for the adaptive sampling strategy. They developed the adaptive method to adjust the sensor node sampling frequency based on the input characteristic, which is the gradient of change of air pollutants. A Kalman filter is used to remove the noise from the measurement and tune the sampling frequency based on the difference between the present and previous measurements. The base station is responsible for analysing the difference between sensor readings and assigning a sampling frequency to sensor nodes. In this way, they achieve an energy-efficient solution that provides accurate data measurements.

A further study [123] focusses on optimizing the trade-off between energy and data-quality parameters. The authors introduce a metric to evaluate energy and quality, which models the relationship of sensing, processing and transmitting with quality and energy. This metric represents the dependency between data accuracy and energy consumption. Furthermore, a generic quality and energy adapting system (QEAS) is suggested, which uses this metric, base station scheduling priority and techniques such as batch processing and

adaptive sampling to optimize both energy efficiency and overall quality. The main contributions of this method are a comprehensive data quality and energy model, an adaptive scheduling method and techniques to improve energy efficiency while satisfying the application's quality requirements.

2.4.2 Spatial correlation-based centralized adaptive sampling

There is a high level of similarity between readings of sensor nodes located in close proximity to each other. This indicates the spatial correlation. Because of spatial correlation, not all neighboring nodes need to sample data.

Willet et al. [119] propose a back-casting scheme for clustered wireless sensor networks. Their algorithm has two main steps. In the first step, called the preview step, cluster heads divide the monitoring area into the clusters with non-uniform resolution. Each cluster head then sends its data to the base station, which is in charge of finding the data correlation from the data received. Considering this correlation, the base station sets a new sampling rate and number of active sensor nodes per cluster for the next time period. In the second step, called the refinement step, the base station may activate additional sensors in those clusters where the spatial correlation is low. This is done using a 'backcast' procedure where the base station sends an activation message to the cluster heads residing in the smallest clustering areas generated by the preview phase. Rotating the role of cluster head periodically among the sensor nodes leads to the balanced energy consumption among the sensor nodes. Energy efficiency and accuracy are the main advantages of this method, which comes at the cost of complexity, latency and scalability.

The approach reported in [124] performs the adaptive sampling by leveraging the blue-noise-masking concept. The minimum number of the sensor nodes that should sense the environment is found by taking the signal to noise ratio (SNR) into account. The proposed method is not scalable and considers only single-hop communication. In the approach developed by Dang et al. [125], the monitoring area is modeled as a set of grid points in which sensor nodes perform sampling. The sensor nodes that sample data are assigned based on the sigma point Kalman filter [126]. When new nodes are added to the network, the computation time, to find the next sampling node; increases exponentially. As such, it is not scalable and leads to more delay while maintaining an acceptable level of data quality.

2.4.3 Spatial-temporal correlation-based centralized adaptive sampling

Combining both temporal and spatial correlations can help design more efficient adaptive sampling approaches.

Rahimi et al. [127] propose the use of a nested stratified random sampling (NSRS) policy for adaptive sampling on mobile nodes. This method uses an actuation-enabled robotic sensor which consists of a number of mobile nodes equipped with meteorological sensors. A mobile node moves in the area of interest and collects samples with a maximum amount of design space coverage (area coverage degree). This mobility plan is developed based on the cost, location, and variation of the area parameters. According to the observations, mobile nodes predict the distribution of the sample points. The authors state that their method strikes a balance between the sampling delay and incorporating knowledge of mobile sensor travelling time. Sensor reading dependency on the mobile node location is forgotten, which results in a lower quality of data. The proposed mobility plan is not fine-tuned for greater dynamicity in the environment, which increases demand for energy; however, it can adjust its prediction model to the high dynamicity condition of the environments.

Pursuing a different avenue of inquiry, Dashpande et al. [128] examines using a model-based query approach, which makes a prediction model via a time-varying multivariate Gaussian process. In this method, only a small fraction of the nodes perform data sampling and transfer their readings to the base station. This node predicts the reading of non-sampling sensor nodes by using its correlation-based model. The aim of this model is to find a set of good sampling nodes to curtail approximation errors. Computational overhead and energy cost are the main concerns with this method. Since there is a significant delay in updating prediction models, the sensor node cannot adjust its prediction model to the important environmental changes, which results in lower data quality.

Gedik et al. [129] propose a model-based adaptive sampling method implemented in two steps. The first step concerns cluster forming by selection of cluster heads. Each node is then assigned to a cluster by considering the similarity of sensor node readings and the hop counts. The cluster construction algorithm is executed periodically based on the power level and sensing behavioral changes of the sensor nodes. In the second step, the proposed solution divides the network into sub-clusters, and then selects at least one sensor node from each sub-cluster as a sampler node. Thereafter, only the sampler node samples data and sends it to the base station. Reading of the

Table 2.1: Centralized adaptive sampling methods

	Data quality	Delay	Energy efficiency	Scalability	Prediction method	Correlation
[115][86]	✓	-	-	-	Cusum Test	Temporal
[117]	-	✓	✓	-	Kalman Filter	Temporal
[118]	✓	-	✓	-	Ensemble Kalman Filter	Temporal
[121]	✓	✓	-	✓	-	Temporal
[109]	✓	-	-	-	Kalman Filter	Temporal
[123]	✓	-	✓	✓	-	Temporal
[122]	✓	-	✓	✓	-	Temporal
[119]	✓	✓	✓	-	Backcasting Scheme	Spatial
[124]	✓	-	-	-	Blue Noise Masking	Spatial
[125]	✓	✓	✓	-	Sigma Point Kalman Filter	Spatial
[127]	✓	✓	-	✓	Nested Stratified Random Sampling	Spatial-Temporal
[128]	✓	✓	✓	-	Time Varying Multivariate Gaussian	Spatial-Temporal
[129]	✓	✓	-	✓	-	Spatial-Temporal
[130]	✓	✓	✓	-	Multi Output Gaussian Process	Spatial-Temporal
[131]	✓	-	✓	-	Fisher Information Test	Spatial-Temporal

non-sampling nodes is predicted for each sub-cluster by using a probabilistic model. The trade-off between energy exhaustion and data quality is addressed in this study.

A multi-output Gaussian process is utilized in [130] to schedule sensor nodes and their sampling rates. This approach calculates the spatial correlation among sensor node observations and based on selects active sensor nodes for each time period. The observations of non-active nodes are predicted using a data regression model. The computational and communication cost of this method is high, but it ensures acceptable data quality.

An adaptive sampling strategy was proposed in [131] to achieve energy efficiency and accurate target-tracking in sensor networks. When the uncertainty in the environmental data increases, this method raises target sampling frequency, while with less uncertainty sampling frequency is reduced. In addition, adaptive sampling with sensor selection provides significant reduction in the number of sensors selected in the entire tracking period, as compared to the fixed target sampling case.

Table 2.1 shows a comparison of the aforementioned approaches and their support for various quality of service parameters. Since in the centralized solution, sensor nodes have to transfer their all sampled data to the base station, proposed solutions can not best satisfy energy efficiency requirements. However, having a complete overview of the environmental conditions in the base station leads to more accurate data prediction. Furthermore, by considering the base station as a central decision making node for the whole network limits the scalability of the network. This table shows that centralized solutions are suit-

bale for small scale delay torrent sensor networks. Simple sampling prediction in sensor node level can increase their scalability and energy efficiency.

2.5 Decentralized adaptive sampling

In most of the decentralized sampling methods, sensor nodes and base stations use similar synchronized prediction models. Predictions about future behavior are essentially constructed at each sensor node. In the case of increasing prediction error, sensor nodes increase their sampling rate, update their prediction model accordingly, and send the recent readings to the base station in order to update the prediction model. If the prediction is accurate, sensor nodes perform their sampling and prediction without sending any data to the base station, mainly because the base station utilizes the same prediction model as the sensor nodes. Therefore, decentralized adaptive sampling mitigates the impact of data communication overhead.

Decentralized approaches often require more control operation to manage interaction between sensor nodes, which can lead to higher complexity and control overhead. Data quality has a direct relationship with the accuracy of the prediction model. Decentralized approaches often ensure scalability and robustness and can support high dynamicity and frequent changes in the monitoring area.

2.5.1 Temporal correlation-based decentralized adaptive sampling

The existing literature [132] introduces SPAS as a routing layer scheduled scheme. This method attempts to decrease packet loss and achieve power efficiency by using adaptive sampling. This approach performs scheduling at the network layer on the grounds that this is more flexible, as there is no strict requirement for time slots. In SPAS, a sensor node wakes up periodically based on the transmission rates of its child nodes. Every sensor node tries to optimize its route in order to reduce the active times and the wake-up times of its parent. This approach considers the transmission rate and the sampling rate to be the same. To change the transmission rate, sensor nodes update their routes and broadcast their new transmission rates in the routing maintenance phase. Thereafter, all other nodes in the path originating from the current node to the base station broadcast the transmission changes. When non-child neighboring

nodes use the channel or the node has finished all its transmissions in the current transmission cycle, it switches to the sleep mode. This work addresses a route optimization mechanism and reduces the number of transmission rates and wake up times of the parent node. This approach is energy efficient and scalable, and it minimizes latency (due to optimized routing), however, it suffers from low data quality.

Padhy et al. [133] propose a utility-based data collection protocol for data monitoring. They model temporal correlations as a piecewise linear function and use a predefined confidence threshold to find an appropriate sampling frequency. As this sampling method is decentralized, each sensor node is equipped by a linear regression model to predict sample values. If the predicted values fall outside the predefined confidence interval, the sampling frequency is increased to a predefined maximum. However, if it lies within the predefined confidence interval, the sampling frequency is decreased by a given factor. This frequency can decrease until it reaches a set minimum. In this approach, sensor nodes not participating in the routing perform sampling more often. The simplicity of the prediction model, energy efficiency, and low latency are the main advantages of this approach, which come at the cost of lower data quality and lower scalability. It should be noted that, although data quality is low, it is still acceptable. Some criteria, such as prediction confidence interval, window length, and sampling rates, must be selected carefully in order to achieve high performance.

The approach reported in [87] employs an autoregressive-moving-average (ARMA) to predict sampling and transmission rates. To mitigate the problem of missing important information when using fixed sampling rates, the authors also adjust their parameters based on a user-specified coverage level. Furthermore, by using cross-layer information, they adapt the sampling rates to the topology changes. When the network is stable and there is no major change in the reported data, both sampling and transmission rates are decreased based on the predicted values. With a decreasing sampling rate, this approach skips a large number of samples and tries to predict these skipped samples. If the difference between the predicted values and the actual values exceeds the predefined error threshold, the sampling rate is changed, and the model is updated. This approach works only in situations where the environment is stable and there are few outliers. Simplicity, energy-efficiency, and low latency are the main advantages of this approach, while it suffers from predictions of low accuracy.

Similar to [87], article [134] uses the Box-Jenkins method for time series

modeling to address adaptive sampling. This approach forecasts future samples based on existing samples and estimates those that can be skipped. Finding an acceptable level of sampling accuracy is one of the challenges addressed here. Similar to the previous approach, this solution is simple and produces low latency at the expense of lower data quality.

Jain et al. [135] propose a dual Kalman filter-based sampling method, in which each sensor node has one version of Kalman filter and tries to perform linear prediction on the smoothed sensor readings. Thereafter, sensor nodes compare the prediction value with the real data and calculate the difference. If this difference is unacceptable with respect to a predefined threshold, the sensor node sends the new readings to the base station. The base station has a specific Kalman filter for each sensor node and, based on the reported data, predicts and constructs the sensor node's behavior. The computational complexity of the prediction model is the main disadvantage, leading to the more energy consumption and delays. However, it guarantees a high degree of data quality.

Khu et al. [136] propose three decentralized sampling methods, which employ Fisher information and a Gaussian process regression to create a prediction model. First, they use the Gaussian process regression to find a temporal correlation among sensor readings and then define sampling rates for each time period. The mean Fisher information over the entire interval is then calculated to measure the content of real samples. Three decentralized algorithms are proposed to find a trade-off between computational cost and optimal sampling strategy. Each sensor node acts as an autonomous agent. In this approach, each node indicates how and when it performs sampling and makes a decision about its individual actions based on its observations and local knowledge. By considering its limited resources, including energy, each node varies its sampling rate in order to maximize the value of information it produces.

An adaptive sampling method is suggested in [137] for a body area sensor network. This solution employs Fisher test theory to predict and tune the sensing frequency. To do so, the sensor node first calculates the test results and residual of the variance test. Based on this result, if the sensor node experiences high variance, it increases its sampling frequency in order to prevent skipping important data, and reduces it when the variance is lower than the threshold. This solution mainly focusses on energy efficiency, while data quality is defined mostly based on the event detection accuracy.

The authors in [85] address a time-series-based data-gathering solution to reduce data redundancy and avoid unnecessary expenditure of power. This

approach changes sampling frequency when significant changes are detected in an environmental parameter. This detection strategy is based on an extension to Holt's method and statistical model. Furthermore, TCP Reno congestion control is utilized to adjust sampling interval parameters in terms of sampling step size and maximum sampling value. Inspired by the additive-increase multiplicative-decrease (AIMD) congestion control mechanism, this approach doubles the sampling step size at each time interval until it reaches the pre-specified maximum step size. It then fixes the sampling frequency at that time interval as long as the data prediction error is below the specified error threshold and no events are detected. During the additive increase phase and the maximum step size maintenance phase, if the prediction error crosses the threshold, the sampling step size is halved.

2.5.2 Spatial correlation-based decentralized adaptive sampling

Santini and Romer [138] propose an adaptive sampling strategy for a quality-based data reduction that improves the approach proposed in [128]. It employs a dual prediction scheme using a least mean squares (LMS) filter without requiring any prior knowledge about the statistical properties of the phenomena of interest. Both sensor nodes and the base station must agree on the predefined model. This process leads to additional overhead. This approach is quite simple, and its computational complexity and memory usage are very low. However, it ensures that the original data can be re-constructed at the base station within a predefined accuracy bound.

In [139, 140], the authors use a linear minimum-variance unbiased model to find a subset of sensor nodes to sample the data. If the environmental conditions are stable and the noise level is low, a minimum number of sensor nodes sample the data. As the noise level increases, more sensors begin sampling. This approach is scalable, energy efficient, and has low latency, while data quality is guaranteed.

2.5.3 Spatial-temporal correlation-based decentralized adaptive sampling

The authors of [141] use data-gathering protocols to achieve energy efficiency and data consistency, turning the sampling rate into the rate of data changes.

Data dynamics describes the data variability by taking into account environmental changes. This protocol adapts the sampling rate by using reinforced learning. Data dynamics are modeled with windowing concepts to store and observe data readings. Two parameters are defined: the number of readings in a window and the number of windows in an observation. Any application can predefine bounds for these two parameters. As this approach is decentralized, each node calculates its data changes and sampling rate. The approach adapts the sampling rate based on a model describing frequency of data dynamics and resource availability. The complexity of this approach is high, but it is energy efficient and provides an accurate data prediction model.

Another study [142] introduces a combination of adaptive sampling and sensor node scheduling to achieve energy-efficient data collection. The spatial-temporal correlation of inter- and intra-sensor nodes observations is exploited to select active sensor nodes for each time period. Furthermore, this method tracks the temporal variation of environmental data and analyses the cause and direction of propagation of the variation in the environment. To decide whether to play the role of a representative active sensor node, each sensor node employs a variety of parameter correlations, Hellinger distance (distance between probability distributions), histogram distance, and vector norms. Active sensor nodes use temporal correlation to tune their data sampling and transmission rate. Non-active sensor nodes wake up adaptively, track the spatial variation of data over time and update their status accordingly to enable accurate data prediction at the base station. Adaptive sensor node scheduling and data sampling reduce the need for power, while tracking the variation in data accuracy ensures that data-quality requirements are met.

A frequency-based adaptive sampling (FAS) algorithm is proposed in [143] that dynamically tunes the sampling frequency of sensor nodes based on the signal frequency change. It measures the periodic signal frequency in different clusters and then adjusts the sampling frequency of sensor nodes according to the requirements of the physical phenomena under observation. As a result, this method meets two complementary requirements of the network: it satisfies the data accuracy requirement by providing a set minimal sampling rate for data reconstruction, for example, Nyquist sampling frequency, and it minimizes energy consumption by reducing sensor node sampling rates.

Table 2.2 summarizes characteristics of aforementioned approaches and their support for various quality of service parameters. In general, the decentralized solutions localize the data sampling adjustment algorithms. In most

Table 2.2: Decentralized adaptive sampling methods

	Data quality	Delay	Energy efficiency	Scalability	Prediction method	Correlation
[132]	-	✓	✓	-	-	Temporal
[133]	✓	✓	-	✓	Piecewise Linear Regression	Temporal
[136]	✓	✓	✓	✓	Fisher information and Gaussian Process Regression	Temporal
[87]	-	✓	✓	✓	-	Temporal
[134]	-	✓	✓	✓	Box-Jenkins	Temporal
[135]	✓	✓	-	✓	Dual Kalman Filter	Temporal
[137]	✓	✓	✓	-	-	Temporal
[85]	-	-	✓	-	Time Series Prediction	Temporal
[138]	✓	✓	✓	✓	Least Mean Squares Filter	Spatial
[139][140]	-	✓	-	✓	Linear Minimum-Variance Unbiased Model	Spatial
[141]	✓	-	-	✓	-	Spatial-Temporal
[142]	✓	✓	✓	✓	-	Spatial-Temporal
[143]	✓	✓	✓	✓	-	Spatial-Temporal

of those solutions, base station and sensor nodes are equipped with the similar data prediction models. Therefore, the predicted data in the base station and sensor nodes are the same which minimizes data transmission necessity. Since the base station predicts sensor node data, it can report data to the end user with minimum delay. Furthermore, as you can see in the table, this localization improves the scalability and data quality of proposed solutions. For some scenarios, decentralized solutions suffer from extra communication overhead among sensor nodes.

2.6 Hybrid adaptive sampling

Designing an energy-efficient and accurate decentralized adaptive sampling method is a difficult task because it imposes high amount of overhead in terms of communication and complexity. On the other hand, although centralized methods are usually energy efficient and accurate, they have scalability and latency problems. Hybrid methods aim to maximize the advantages of both techniques.

The self-organizing resource allocation method [144] suggests a market-based resource allocation strategy for the sensor network. Sensor nodes can sell some of their parameters, such as sampling rate, according to the global information published from the base station. According to the current status of the monitored environment, the base station defines prices for various actions of each node. When the environmental conditions are stable, it increases the price of data listening and relaying parameters. The key point of this approach is that

it takes the relationship between network parameters and energy expenditure into consideration. In this marketing game, sensor nodes try to maximize their utility by considering their energy resources, for which temporal correlation is used. On the other hand, the base station attempts to set appropriate prices in order to maximize the global network utility, for which spatial correlation is used. This approach guarantees quality of service in terms of latency and accuracy, but computational complexity and energy consumption are high.

The authors of [145] propose an efficient and novel approach to determine the number and locations of sampler nodes, which limits the prediction error for both stationary and mobile sensor networks. A Gaussian process allows for predictions of data at any unobserved locations. The first part of this research addresses a simple and accurate solution for the sensor node selection problem. The second part of the research introduces a hybrid spatial-temporal method for data prediction modeling in a mobile robotic wireless sensor network. The proposed sampling policy focusses on minimizing the error in all unobserved areas to obtain the most informative data in each time period. This minimization is addressed by a greedy algorithm in polynomial time. The computational complexity of the solution is improved using a Gaussian Markov random field (GMRF), such that the computational complexity of the greedy algorithm to solve the resulting optimization is deterministic even with an increasing number of measurements.

A further study [146] introduces an energy-aware adaptive sampling approach. This method is a combination of an adaptive sampling algorithm and a power-management technique tailored to energy-harvesting sensor networks. Each sensor node updates its sampling frequency based on the current energy level of the network. When the battery level is higher than a user-defined threshold, sensor nodes tune their sampling rate using a sampling frequency defined by the base station. However, when the battery power drops below the threshold, the algorithm becomes more energy-conservative by minimizing the sampling rate. Thus, the sensor node manages its sampling frequency based on its power level. The combination of energy-harvesting, an adaptive sampling algorithm and an energy-aware management technique provides a self-sustainable wireless sensor network solution, but data accuracy is not ideal, providing only minimal accuracy.

A summary of the aforementioned approaches and their impacts on various quality of service parameters is represented in Table 2.3. Combining centralized and decentralized solutions helps hybrid methods to propose localized sampling scenarios and more accurate overview about environment con-

Table 2.3: Hybrid adaptive sampling methods

	Data quality	Delay	Energy efficiency	Scalability	Prediction method	Correlation
[146]	✓	-	✓	-	-	Temporal
[144]	✓	-	-	✓	-	Temporal
[145]	✓	-	✓	✓	Gaussian processes	Spatial-Temporal

dition in the base station. Therefore, energy efficiency and data quality requirements are mostly met. However, hybrid solutions suffer from complexity which should be improved.

2.7 Chapter summery

This chapter has reviewed and compared adaptive sampling approaches and their support for various quality of service parameters. To this end, we divided adaptive sampling approaches into centralized, decentralized and hybrid categories. For the centralized and decentralized methods, we categorize the existing methods into temporal, spatial and spatial-temporal correlation-based approaches. Our comparison shows that none of the existing adaptive sampling approaches satisfy all quality of service parameters, and no studies have yet considered how to strike the optimal balance between all these parameters. The centralized adaptive sampling solutions usually require a sampling adjustment mechanism in base station to calibrate sensor nodes' sampling frequency. This category of adaptive sampling approaches might not give the best performance in large-scale sensor networks. In case of decentralized adaptive sampling solutions, the base station and sensor nodes usually utilize the same data prediction and sampling adjustment procedure. Therefore, the sensor nodes can locally decide their sampling frequency and in case of inaccurate data prediction, report data to the base station. This localization best suits the large-scale sensor network and minimizes data transmission costs. While proposed adaptive sampling solutions answer many research questions raised in this area, there are other research issues which are not well addressed yet. A comprehensive investigation on the effects of statistical data prediction models on energy efficiency and data accuracy parameters are one of those open research issues. Data accuracy, prediction model complexity and frequency of sampling rate adjustment are the essential parameters of data prediction model which influence network performance.

Spatial-Temporal Correlation-based Adaptive Sampling

3.1 Introduction

Environmental data, for example temperature, usually changes gradually. It often follows a normal distribution and falls into a specific range. As long as data falls within such a given range, its behavior is assumed to be normal. Existing adaptive sampling techniques, however, often ignore this property and aim to identify any small changes in the data to meet the data-quality requirements of the application. As such, they deplete power in transmitting irrelevant data.

In this chapter, we propose an adaptive sampling approach that leverages the benefit of this concept to achieve a trade-off between energy efficiency and data quality. While changes in sensor readings that fall within a given data range are not reported, those closer to the boundaries of the data range are expected to have more values. In other words, those changes may indicate that the future readings are quite likely to fall outside the range and as such need to be transmitted.

With this in mind, we take advantage of spatial-temporal correlation among sensor nodes and their readings to provide energy-efficient scheduling for sensor nodes and to adjust the sampling frequency for each node. The main idea behind this approach is to use a carefully selected, dynamically changing subset of sensor nodes to sense and report data. Furthermore, each sampling node tunes its sampling frequency based on the environmental conditions, and uses spatial-temporal correlation-based prediction models to estimate non-sampler nodes' readings, as well as their skipped samples.

This chapter makes the following contributions:

- **Sub-clustering and sensor node selection:** A new spatial correlation-based active sensor node selection method is proposed. A K-means sub-clustering algorithm is presented that uses the spatial correlation between neighboring sensor nodes in order to group them into different sub-clusters. Within each sub-cluster, representatives are chosen for the sampling task. Sensor nodes located in the same sub-cluster exploit the similarity between their readings to select the sensor node that can best predict their readings. The proposed solution diminishes the number of active sensor nodes while ensuring accurate data prediction for non-active sensor nodes.
- **Auto-regressive moving average model-based data gathering:** Central to the proposed solution is an ARMA based data modelling technique that addresses energy-efficient and accurate data gathering. The cluster heads build a data prediction model that captures temporal correlation of sensor readings of each sensor node. Each sensor node compares its readings with the predicted data of this model. It pushes its data back to the base station only when the difference between the real measurement and the predicted data is more than a predefined error threshold. This model restricts the data transmission to only reporting data that is not predicted accurately, thereby ensuring data quality.
- **Adaptation of sampling frequency to environment dynamics:** We adapt the sampling frequency based on the dynamic changes of the environment. As long as stable conditions are observed, sensor nodes skip more samples. When conditions are unstable, sensor nodes sample with maximum frequency. In this way, our approach utilizes a dynamic sampling frequency adaptation to the environmental dynamics.

3.2 Preliminaries

Before we explain our adaptive sampling method, this section explains the network model and introduces some basic definitions and concepts used throughout this chapter.

3.2.1 Network architecture

We consider a two-tier network architecture for the data collection. It consists of sensor nodes which collect data at various sampling frequencies and more

Figure 3.1: Network architecture

powerful nodes acting as cluster heads. It is assumed that all sensor nodes are homogeneous in terms of their sensor type and their packet size. This architecture is illustrated in Figure 3.1.

Each cluster head is responsible for collecting data from its cluster nodes and sending it back to the base station. The cluster heads use single-hop communication with their cluster sensor nodes and the base station. There may be multiple cluster heads distributed in the environment, and each cluster head manages tens of sensor nodes. The clustering is performed by the cluster heads based on the similarity between the readings of the sensor nodes.

3.2.2 Energy model

There is a specific resource cost associated with each sampling strategy. Energy is the scarcest resource in wireless sensor networks. Energy consumption is directly related to (i) the number of active sensor nodes in the sampling process and (ii) the sampling frequency of the active nodes. In order to find this cost, we use the energy model in [147].

To calculate total energy consumption, it is necessary to determine the energy consumption of each operation performed by the sensor nodes and the

cluster head during each time interval. The energy required by each sensor node and the cluster head is related to five basic tasks: processing, radio transmission and receiving, transient energy, sensor sensing and sensor logging. With cluster heads, the total number of data packets that each receives and processes is the sum of the data packets sent by the sampling nodes of that cluster. As mentioned before, we assume that packet size and transmission distance are fixed for all sensor nodes and all sensor nodes (except the cluster head) are homogeneous. Therefore, the energy requirement of sampling tasks for all sensor nodes is the same. In a spatial approach, it is assumed that sensor nodes only sense and send data to their cluster head, while in a temporal approach, sensor nodes also perform some computations on the sensor data.

The energy consumption of sensor nodes and the cluster head can be calculated as follows:

$$E_{node_i} = E_{sens} + E_{log} + E_{proc} + E_{rtx}, \quad (3.1)$$

$$E_{CH_j} = E_{rx} + E_{log} + E_{proc} + E_{tx}, \quad (3.2)$$

,where E_{sens} denotes the energy cost of sensing; E_{rtx} denotes the energy cost for transmitting one packet and receiving control messages from the cluster head; and E_{log} and E_{proc} denote the energy costs for logging and processing data on a sensor node, respectively. In the case of the cluster head, E_{rx} and E_{tx} denote the energy costs for receiving one packet, and E_{log} and E_{proc} denote the energy costs for logging and processing sensor nodes' readings on the cluster head.

3.2.3 Data quality metric

In order to quantify data quality precisely, we evaluate the prediction accuracy of unavailable samples using our spatial-temporal data correlation model. Current adaptive sampling approaches attempt to track every change in data readings to satisfy data-quality requirements. To do so, they compare the predicted values with the observed data using the following equation:

$$Err(X_t) = |X_t - R_t| \quad (3.3)$$

,where X_t and R_t denote the predicted value and the real observation at time t , and Err_t is the prediction error.

Table 3.1: Region definition

Real data	Predicted data	Prediction accuracy
$R < Lower \parallel R > Upper$	$Lower \leq X \leq Upper$	Not Accurate
$R < Lower \parallel R > Upper$	$X < Lower \parallel X > Upper$	Accurate
$Lower \leq R \leq Upper$	$X < Lower \parallel X > Upper$	Not Accurate
$Lower \leq R \leq Upper$	$Lower \leq X \leq Upper$	Accurate

If this error exceeds a predefined threshold, the environmental condition cannot be well covered by the current sampling rate or nodes and an adaptation is needed.

The fact that certain environmental data can be considered as irrelevant, gives us a reason to employ a different data quality metric. Therefore, we introduce an application-dependent range denoted by $[Lower, Upper]$. In our approach, each sensor node evaluates its prediction as shown in Table 3.1. According to this table, when a sensor node correctly predicts that data is either within $[Lower, Upper]$ or out of that range, its prediction is accurate; otherwise, the prediction is inaccurate.

3.2.4 Region definition

As seen in Figure 3.2, by considering the predefined data range of $[Lower, Upper]$, sensor node readings may fall in three different regions, which we call A , B , and C .

Region A represents in-bound readings, while regions B and C represent out-of-bound readings. As sensor node readings become closer to the boundaries of each region, they are more likely to fall outside that boundary and within other boundaries. Thus, these readings are more critical to report. Therefore, the importance of a sensor node reading depends on how close its value is to the range boundaries.

To measure the proximity of sensor node readings to the boundaries and to find the proper weighted value, a metric must be developed. To this end, we use the difference between the borders of the data range denoted by ($Upper$ or $Lower$) and the mean of a given range denoted by ($Middle$) as the maximum difference (Max_Diff) denoted by ($Upper - Middle$ or $Lower - Middle$).

As sensor node readings approach the region boundaries, the difference between sensor node readings (R) and the mean of a given range ($Middle$) decreases. Based on the Max_Diff parameter, we introduce a new coefficient

Figure 3.2: Data region definition

denoted by d , which is the determining parameter in tuning sensor nodes' sampling frequency and selecting sampler nodes. This d parameter is defined as follows:

$$d = \frac{R - Middle}{Max_Diff} \quad (3.4)$$

The value of d varies from -2 to 2 , based on which we characterize each region as shown in Figure 3.2.

3.3 Overview of proposed approach

Our approach can be divided into four steps, i.e., data prediction model building, sub-clustering, sampler node selection and data sampling. These are described in more detail below

- **Data prediction model building:** In order to lower down the number of data transmissions between the sensor node and the cluster head, we use a data prediction model. The cluster head gathers sensor node readings and on this basis creates a data prediction model using the ARMA model[148]. It then sends the model and its parameters to the sensor nodes. Because the cluster head and sensor node have the same prediction model, the readings which are predicted accurately are not pushed to the cluster head.
- **Sub-clustering:** The sub-clustering must be implemented in each cluster. To this end, the cluster head first collects non-periodic samples (in forced sampling times) from all sensor nodes to determine the spatial correlation among sensor nodes' readings. The cluster head then employs a centralized K-means clustering approach which executes every t_{SC} times in order to form sub-clusters inside the given cluster.
- **Sampler node selection:** Sampler node selection is performed in three steps: first, each sensor node compares the average of its weighted value W with other nodes to determine the interest level (IL) parameter for each neighbor. Interest level is a quantity metric showing the extent to which the given sensor node is interested in choosing a specific neighbor node as a sampler node. In the second step, sensor nodes exchange their interest levels. Finally, each sensor node takes into account the interest of other nodes in itself, its energy level and the data-quality metric to calculate its ability to act as a sampler node. It then broadcasts its sampler capability level to other nodes. The node with the highest interest level is selected as a sampler node.
- **Adaptive data sampling:** After sub-clustering and selecting sampler nodes, we define the sampling period as the duration of time when sensor node must collect data from the environment. For collecting data, two different sampling policies for sampler and non-sampler nodes are suggested.
 1. **Temporal correlation-based adaptive sampling:** Sampler nodes conform to a temporal correlation-based adaptive sampling approach to gather data. In this mechanism, sampler nodes employ a time series prediction model to adjust their sampling frequency and predict the skipped samples.
 2. **Forced sampling:** In our approach, only sampler nodes report data to the cluster head, while it requires all sensor nodes' observations to

organize the sub-clusters. Therefore, we introduce periodic forced sampling times at which the non-sampler nodes wake up and perform sampling.

A general overview of this approach is shown in Figure 3.3.

3.4 An energy-efficient adaptive sampling approach

Our adaptive sampling approach has the following four major steps: (i) sub-clustering, (ii) building the data prediction model, (iii) sampler node selection

Figure 3.3: Overview of proposed approach

and (iv) decentralized temporal correlation-based adaptive sampling. Steps (i) and (ii) are executed on the cluster heads, while steps (iii) and (iv) are executed on the sensor nodes. Each of these steps is explained in more detail in the following sections.

3.4.1 Sub-clustering

The first step in our approach is sub-clustering, during which the cluster head collects periodic samples (at forced sampling times) from all sensor nodes to determine the spatial correlation among the sensor nodes' readings. The cluster head organizes each cluster into sub-clusters in three steps: (i) data gathering, (ii) pre-processing and (iii) K-means clustering.

3.4.1.1 Data gathering

Forced sampling is the only data-gathering policy which periodically collects data from all sensor nodes. Therefore, the cluster-head employs the forced samples collected in the most recent sampling period to generate a new set of sub-clusters.

During the forced sampling period, the sampler and non-sampler nodes perform sampling at the designated times. We denote the forced samples collected by each sensor node as FD_i , which is a vector of consecutive forced samples gathered by node i . At the end of the sampling period, the sensor nodes calculate the mean ($\mu_i(t)$) and the standard deviation ($\sigma_i(t)$) of their forced samples. Then they send them to the cluster head to be used for the sub-clustering process.

$$\mu_i = \frac{\sum_{j=1}^{NFS} FD_i(j)}{NFS} \quad (3.5)$$

$$\sigma_i = \sqrt{\frac{1}{NFS} \times \sum_{j=1}^{NFS} (FD_i(j) - \mu_i)^2} \quad (3.6)$$

, where NFS is the number of forced samples.

3.4.1.2 Data Preprocessing

Upon receiving the μ_i and σ_i values at time t , from the sensor nodes, the cluster head performs sub-clustering using a K-means clustering algorithm. Instead

of using the raw data as the input for the K-means clustering algorithm, we consider the weighted values to be calculated using Equation 3.7. In this regard, the cluster head first divides sensor nodes into three classes based on their readings. The first class relates to the sensor nodes whose readings are in-bounds (i.e., region *A* in Figure 3.2). The second and third classes relate to the sensor nodes whose readings are out-of-bounds (i.e., region *B* and *C* in Figure 3.2). After classifying sensor nodes into the three categories, the cluster head calculates the weighted value of the sensor nodes readings as follows:

$$\begin{cases} W_i = 0 & \text{if } \mu_i \in B_2 \\ W_i = |\mu_i| \times \frac{\mu_i - Upper}{(2 \times Upper - Middle) - \mu_i} & \text{if } \mu_i \in B_1 \\ W_i = |\mu_i| \times \frac{Upper - \mu_i}{\mu_i - Middle} & \text{if } \mu_i \in A_1 \\ W_i = |\mu_i| \times \frac{\mu_i - Lower}{Middle - \mu_i} & \text{if } \mu_i \in A_2 \\ W_i = |\mu_i| \times \frac{Lower - \mu_i}{\mu_i - (2 \times Lower - Middle)} & \text{if } \mu_i \in C_1 \\ W_i = 0 & \text{if } \mu_i \in C_2 \end{cases} \quad (3.7)$$

The proposed weighted values take two factors into account: first, the mean of sensor node readings, and second, the closeness of that mean to the boundaries. According to Figure 3.2, if there are *C*2 and *B*2 regions whose readings are too far from the lower and upper bounds, respectively, their values are assumed to be zero since their readings are no longer critical.

3.4.1.3 K-Means clustering

After pre-processing, the cluster head with the weighted values runs K-means for each region. For the sensor nodes whose readings are in region *A*, K-means is requested to group sensor nodes into three different sub-clusters whose centres are near Middle, Lower and Upper. For the sensors whose readings are located in *B* or *C* regions, two sub-clusters are sufficient. As we mentioned in the previous section, in regions *B* and *C*, critical readings belong to the sensor nodes whose readings are close to Upper and Lower, respectively. Other readings can be considered as further readings, which are not critical. Therefore, for region *B*, two sub-clusters with centres close to Upper and $(2 \times Upper - Middle)$ are required. Similarly, for region *C*, two sub-clusters with centres near Lower and $(2 \times Lower - Middle)$ are considered. We use a general K-means clustering technique for sub-clustering [149].

The cluster-head analyses the resulting sub-clusters to see whether they can be merged in order to have fewer clusters. This requires a similarity metric, for which we use the difference between the mean of the two sub-clusters. If this difference is less than the minimum distance between the mean of the sub-clusters, then they can be merged. The minimum difference can be calculated as follows:

$$|M_i - M_j| \leq [(1 - \max(d_i - d_j) \times \text{Max_Diff})] \quad (3.8)$$

, where M_i and M_j represent the centre of sub-clusters i and j ; and d_i and d_j represent the closeness of the centre of sub-clusters i and j to the closest region boundaries, respectively.

The sub-clustering flowchart is represented in Figure 3.4.

3.4.2 Building the data prediction model

In order to minimize the number of data transmissions between the sensor nodes and the cluster head, we apply a data prediction model to forecast upcoming readings. To do so, the cluster head builds an ARMA-based prediction model. This model is a mathematical representation of the auto-correlation in a time series. An ARMA model has two parameters, p and q , which refer to the order of auto-regressive (AR) and moving average (MA) parts of the model, respectively. The AR parameter p represents the temporal correlation in the time series and is used to predict future readings based on a linear weighted sum of past readings, while the MA parameter q represents the relationship between estimation errors and is modelled as a zero-mean white noise [150].

The key assumption for the model is that physical parameter monitored by the sensor node represents a short- or long-term correlation, and previous readings can be used to predict future readings. The cluster head then sends the model and its parameters to the sensor nodes. Having the same prediction model on the cluster head and the sensor nodes implies that the sensor readings that are accurately predicted will not be transmitted to the cluster head.

The complete $ARMA(p, q)$ is defined as follows:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \dots + \theta_q Z_{t-q} \quad (3.9)$$

, where $Z_t \sim WN(0, \sigma^2)$ and $\phi_i, i = 1, 2, \dots, p$ and $\theta_i, i = 1, 2, \dots, q$.

Because of the resource restriction of the sensor nodes, we select an ARMA with order $(1, 0)$, which has low complexity. Using this model, the cluster head

Figure 3.4: Sub-clustering procedure

gathers the sensor node readings and employs the least squares method to predict the model parameters on a per-sensor basis (see [150] for the detailed procedure for estimating these parameters). The constant can now be predicted using the Yule-Walker estimator.

Only when the difference between the real value and the prediction is higher than a threshold, the sensor nodes send their readings to the cluster head. Therefore, the real values of the previous readings may not be known to the cluster head. Lack of real data transmission indicates that the prediction model is still accurate, and the cluster head can simply draw upon the value from the prediction model as an approximation of the real value.

Both the cluster head and the sensor nodes use the ARMA model to estimate sampled values. At the cluster head, the predictions are considered a substitute for the real sensor node readings, while at the sensor node level it is used to measure the accuracy of prediction model. Finally, it is worth noting the asymmetric property of our model. The initial model identification and parameter estimation are burdensome tasks implemented on the cluster head, while predicting the sensor node readings using the proposed estimation model is a simple task implemented on both the sensor node and the cluster head.

3.4.3 Sampler node selection

The sampler node selection procedure aims to determine a set of sensor nodes to sense and sample the phenomena of interest.

When the sub-clustering step is finalized, the cluster-head broadcasts a small control packet by which it informs sensor nodes about their sub-clusters and their sub-cluster head. Upon receiving this packet, the sensor nodes execute this step to select the best candidate nodes as the sampler nodes for the next sampling period. This process is performed in a distributed fashion.

In order to select the proper sampler nodes, sensor nodes within each sub-cluster first exchange the weighted mean and the standard deviation of their last forced samples among themselves and with the cluster head. Upon receiving this data, sensor nodes find their interest level, denoted by IL , in the other sensor nodes. The interest level IL_{ij} indicates degree of interest node i has in selecting sensor node j as a sampler node for its sub-cluster.

We define IL_{ij} based on Equation 3.10 with two terms. The first term represents the similarity between the weighted readings of sensor nodes i, j and is expressed as the normalized difference between the weighted mean values. The second term is the weighted deviation of node i , normalized by the weighted maximum standard deviation of the sub-cluster. This term provides insight into the stability of the sensor node readings.

$$IL_{ij} = (1 - \frac{W_i - W_j}{W_i - W_{Far}}) \times (1 - \frac{\sigma_j}{\sigma_{max}} \times \frac{d_j}{d_{max}}) \quad (3.10)$$

In order to reduce the computation cost imposed by sensor nodes having little or no correlation, the similarity parameter is calculated only for the sensor nodes which meet the following conditions:

$$1. \quad |\mu_i - \mu_j| \leq (1 - \text{Max}(d_i, d_j)) \times \text{Max_err}_{th} \quad (3.11)$$

$$2. \quad \begin{cases} (\mu_i \pm \sigma_i) \in A & \text{if } i \in A \\ (\mu_i \pm \sigma_i) \in B & \text{if } i \in B \\ (\mu_i \pm \sigma_i) \in C & \text{if } i \in C \end{cases}$$

These conditions state that two sensor nodes' readings are similar if the difference between their readings is less than a pre-defined error threshold and the readings after their deviations remain within pre-specified regions.

A proper sampler node is a node that (i) has readings similar enough to the readings of the sub-cluster members, (ii) has enough energy and (iii) covers a large number of sensor nodes. With these conditions in mind, the final decision regarding which sensor nodes will act as sampler nodes is made using Equation 3.12. The first term in this equation shows the sum of the interest levels of sub-cluster members, while the second term represents the number of sub-cluster nodes that can be covered by this sampler node. This parameter ($N_{covered}$) is obtained based on the number of sampler nodes whose interest levels are received by sampler node j . Energy (E_j) indicates the available energy for sensor node j to act as a sampler. This parameter is derived from the current energy level of the node normalized by the maximum energy level of that node. The last term indicates how well the measurements of a given sampler node can represent the measurements of the sub-cluster. This metric is defined on the basis of the difference between the weighted values of the sampler node and the mean of the sub-cluster readings.

$$SD_j = \sum_{i \in cl_k} IL_i \times N_{covered} \times \frac{E_j}{E_{max}} \times \frac{1}{W_j - W_{mean}} \quad (3.12)$$

When sensor nodes find their SD level, they broadcast it to the other sub-cluster members. Thereafter, every sensor node selects the node with the maximum SD value as its sampler node. The sensor nodes that are not interested

in the selected sampler node must find another sampler node by repeating the same selection procedure.

3.4.3.1 Update sensor node roles

During a sampling period, it is quite likely that correlations among sensor nodes will change and that the sampling node will not compensate for non-sampling nodes. Therefore, non-sampling and sampling nodes must be able to update their correlations, which may change the role of sampling and non-sampling nodes. The updating procedure can be implemented only during the forced sampling times, when all sensor nodes are awake. At these time stamps, all sensor nodes perform sampling and compare the similarity of their current readings with past readings, and based on this similarity update their status to act as sampling or non-sampling nodes. In fact, forced sampling times provide the opportunity for sensor nodes to check their similarity with other nodes and update their status if needed.

This opportunity may lead to one of the following two situations:

- If the sampling nodes' recent readings fluctuate or if no sensor node is interested in selecting this node as its sampling node, it changes its status to a stand-alone sampling node. In this state, the sampling node is responsible only for its own readings. It then broadcasts a leave message to its neighboring nodes and informs them to change their sampling node. Upon receiving this leave message, non-sampling nodes perform the distributed sampling selection algorithm to find a new sampling node.
- If the sampling node readings are stable, the sampling node sends a control message containing the average and standard deviation of its readings to its neighboring nodes. Upon receiving this message, non-sampling nodes compare their forced samples with the readings of the sampling node. If their readings are still correlated and no other sensor sampling node has higher SD values, the non-sampling node does not change its status; otherwise, it stays awake and sends a sampling selection message. If more than one non-sampling node attempts to change its sampling status, they run the sampling node selection procedure; otherwise, the non-sampling node announces itself as a standalone sampling node.

Figure 3.5 illustrates sampler node selection procedure.

Figure 3.5: Sampler node selection procedure

3.4.4 Temporal correlation-based sampling rate adjustment

The adaptive sampling rate mechanism presented here has a decentralized sampling policy in which each sensor node decides its own sampling rate. When a sensor node experiences stability in the environmental conditions, it reduces its sampling frequency. In this way, the number of data transmissions between the sensor node and the cluster head is also reduced. Both sensor nodes and the cluster-heads employ the same prediction model, namely ARMA, which describes the given environmental attributes well. Utilizing ARMA, each sensor node transfers data to the cluster head only when environmental conditions are not stable and its data prediction is therefore not accurate enough.

After being deployed, sensor nodes receive the error threshold level that can be tolerated by the application from the cluster head. Each sensor node then initially gathers the first r consecutive sensor readings and stores them in a buffer. Each sensor node also sends these readings to the cluster head, which exploits the past data to predict the future data for its cluster members. Having r last readings, the readings for the epoch $(r + 1)$ are not only acquired but also predicted. The sensor node adopts the ARMA model to predict the $(r + 1)^{th}$ reading and then compares it with its real reading. We refer to the n^{th} reading acquired after r readings in the buffer as R , and the n^{th} reading predicted after r readings in the buffer as X .

If the predictions made using the ARMA model are accurate enough, the sensor node assumes that the prediction approach can be used for the next epoch. This case, as shown in Table 3.1, happens when the predicted value and the reading of the sensor node are either in or out of bounds. In this case, the sensor node skips the next readings. To know how many samples need to be skipped, we define Last Skip Sample (LSS). In order to track the number of skipped sample, another variable Skip Sample (SS) is defined and set to the value of LSS . For each skipped sample, the value of SS is decreased by 1. When SS reaches zero, the sensor node samples another reading and compares it with its prediction. If the prediction is accurate, LSS is increased by 1 and SS is set to the new value of LSS .

This process continues as long as predictions are accurate. When predictions become inaccurate, the sensor node must sample at each time stamp. To do so, LSS and SS are set to zero and the sensor node senses until an accurate prediction is obtained. If the prediction is inaccurate, the sensor node sends the reading to the cluster head in order to update the prediction model.

Notably, if there is a long series of correct predictions, LSS is increased

infinitely. To address this issue, a Maximum Skip Sample Limit ($MSSL$) is defined to limit the maximum number of samples that can be skipped. If LSS reaches $MSSL$, it remains unchanged even if further correct predictions are made. The maximum value that can be assigned to $MSSL$ is 2 less than the buffer size ($Buf_size - 2$).

We adjust $MSSL$ based on the environmental conditions and the sensor node readings. This solution enables sensor nodes to adjust their $MSSL$ parameter autonomously. In addition, it can detect events of any duration. However, there are three different possibilities to adjust this parameter. In what follows, we explain all those possibilities.

3.4.4.1 Adjustment possibilities

In order to adjust $MSSL$, we use the d parameter introduced in Section 3.2.4 to show the ratio of differences between sensor node readings and the maximum difference. This coefficient is used by the maximum allowed value of $MSSL$ ($Buf_size - 2$) to adjust $MSSL$.

$$d = \frac{|data - mean|}{|a - mean|} \quad (3.13)$$

Considering this parameter, Maximum Skip Samples Limit $MSSL$ can be adjusted to the environmental conditions in three different ways. In what follows we explain each of those three ways to adjust $MSSL$ parameter:

1. This adjustment is illustrated in Figure 3.6 for different data regions. In this alternative, we totally adjust $MSSL$ value dynamically and it changes based on the d value changes as follows:

$$\begin{cases} \text{if } d \leq 1 & MSSL = (Buf_size - 2) \times ((1 - d)) \\ \text{if } 1 < d \leq 2 & MSSL = (Buf_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buf_size - 2 \end{cases}$$

In region A , d varies between zero and 1 ($0 \leq d \leq 1$) based on the reading distance to the mean. In this case, $MSSL$ is adjusted as follows:

$$MSSL = \lceil (Buf_size - 2) \times (1 - d) \rceil \quad (3.14)$$

In region B , readings vary between $[a - mean]$ and $2 \times [a - mean]$ or between $[b - mean]$ and $2 \times [b - mean]$, therefore, $1 < d \leq 2$. Accordingly, $MSSL$ is adjusted as follows:

Figure 3.6: $MSSL$ value adjustment for the first alternative

$$MSSL = \lceil (Buff_size - 2) \times (d - 1) \rceil \quad (3.15)$$

This means that once the sensor node reading is close to the boundary, $MSSL$ takes small values, while sensor nodes far from the boundary use large $MSSL$ values.

In region C , we use a fixed value for $MSSL$. In this case, sensor node reading is far from the boundaries, so $MSSL$ can be assigned the maximum number of skips.

$$MSSL = Buff_size - 2 \quad (3.16)$$

2. The second adjustment alternative is illustrated in Figure 3.7 and defined as follows:

Figure 3.7: *MSSL* value adjustment for the second alternative

$$\begin{cases}
 \text{if } d \leq CRe & MSSL = Buff_size - 2 \\
 \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2) \times (1 - d) \\
 \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\
 \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\
 \text{if } d > 2 & MSSL = Buff_size - 2
 \end{cases}$$

In region A, we define two subregions based on two other parameters CRe and FRe which show whether the sensor readings are too close to the boundary or to the mean, respectively. These two parameters can be considered as a fraction of d value. Now, we explain the *MSSL* adjustment procedure for different regions.

In region A, when $d \leq CRe$, the sensor node readings are too far from the mean, so sensor node can sample with the minimum sampling frequency which is equal to $Buff_size - 2$.

$$MSSL = Buff_size - 2 \quad (3.17)$$

When $CRe < d \leq FRe$, we dynamically adjust *MSSL* parameter based on d value as follows:

$$MSSL = (Buf_size - 2) \times (1 - d) \quad (3.18)$$

When $Fre < d \leq (2 - Fre)$, the sensor node readings can fall in region *A* or *B* and they are so close to the boundary, therefore, these readings are considered as critical data. Therefore, we adjust *MSSL* parameter to 1:

$$MSSL = 1 \quad (3.19)$$

In Region *B*, when $(2 - Fre) < d \leq 2$, the sensor node readings can be calculated as follows:

$$MSSL = (Buf_size - 2) \times (d - 1) \quad (3.20)$$

In Region *C*, sensor node readings are too far, therefore, we fix *MSSL* value to the maximum possible value as follows:

$$MSSL = Buf_size - 2 \quad (3.21)$$

3. This adjustment is illustrated in Figure 3.8 and defined as follows:

$$\begin{cases} \text{if } d \leq Cre & MSSL = Buf_size - 2 \\ \text{if } Cre < d \leq Fre & MSSL = (Buf_size - 2)/2 \\ \text{if } Fre < d \leq (2 - Fre) & MSSL = 1 \\ \text{if } (2 - Fre) < d \leq 2 & MSSL = (Buf_size - 2)/2 \\ \text{if } d > 2 & MSSL = Buf_size - 2 \end{cases}$$

In the previous case, the *MSSL* parameter, for some regions, is fixed to the constant values and for the rest it is dynamically updated by *d* value. In this alternative solution, we use another policy and totally fix *MSSL* value into constant values in different regions.

In region *A*, when $d \leq Cre$, the sensor node readings are too far from the mean and the *MSSL* value is fixed as follows:

$$MSSL = Buf_size - 2 \quad (3.22)$$

When $Cre < d \leq Fre$, we adjust *MSSL* parameter based on *d* value as follows:

$$MSSL = (Buf_size - 2)/2 \quad (3.23)$$

Figure 3.8: $MSSL$ value adjustment for the third alternative

When $FRe < d \leq (2 - FRe)$, the sensor node readings can fall in region A or B and we adjust $MSSL$ parameter as follows:

$$MSSL = 1 \quad (3.24)$$

In Region B, when $(2 - FRe) < d \leq 2$ the $MSSL$, value is fixed as follows:

$$MSSL = (Buff_size - 2)/2 \quad (3.25)$$

In Region C, sensor node readings are too far, therefore, we fix $MSSL$ value as follows:

$$MSSL = Buff_size - 2 \quad (3.26)$$

3.5 MSSL adjustment

To achieve an energy efficient and accurate solution, we have selected the first alternative in previous section to adjust *MSSL* parameter. This solution adjusts the *MSSL* parameter dynamically, therefore, sensor node sampling frequency changes according to the environmental condition. On the hand, every changes in the enviroment condition are captured which helps to improve data quality parameter by reporting just informative data as needed. Two other alternatives employ fixed values for *MSSL* parameter. Therefore, it is more likely to miss some unexpected environmental data changes. This reasoning is supported by the experimental results in the simulation section. During the sampling period, it is quite likely that the correlations among the sensor nodes change and the sampler nodes will not represent the phenomena well. To compensate for this effect, forced sampling is required.

Algorithm 3.1 illustrates the decentralized temporal correlation-based adaptive sampling procedure.

3.6 Performance evaluation

In this section, we evaluate our method and compare it with two other approaches. Our evaluation consists of two parts. The first part evaluates the temporal correlation-based sampling frequency adjustment algorithm and compares it with the approach proposed in [87]. In the second part, we compare our spatial-temporal correlation-based algorithm with the similar adaptive sampling approaches addressed in [129].

As we mentioned in Chapter 1, this thesis focuses on temperature monitoring applications. To evaluate our approach, we use publically available temperature dataset provided by Intel research lab in Berkeley [151]. This dataset consists of 54 sensor nodes and represents high degree of spatial and temporal correlation. Furthermore, due to human intervention, there is a lot of randomness in data which increases the variability of data trends. Variability and correlation existed in this dataset are the main characteristics which are needed to evaluate the performance of our solution. In addition, this dataset is used in variety of adaptive sampling research area. The characteristics of sensor node platform used in this dataset is described in Table 3.2.

We assume the data range is $[-0.5, 4]$, the buffer size is 20 and the user defines 90% as the prediction accuracy requirement. The network architecture consists of several single-hop clusters. Since our algorithm runs inside each

Algorithm 3.1 Sampling frequency adjustment

```

1: Get Sensor node readings at every time step till buffer is filled up
2: Transfer data to the cluster head
3: Observe next sample  $R$ 
4: Predict the next sample value  $X$ 
5: if ( $R \in [a, b]$  and  $X \in [a, b]$ ) or ( $R \in [a, b]$  and  $X \in [a, b]$ ) then
6:   if ( $d \leq 1$ ) then
7:      $MSSL = (Buff\_size - 2) - ((Buff\_size - 2) \times d)$ 
8:   else if ( $1 < d \leq 2$ ) then
9:      $MSSL = (Buff\_size - 2) \times (d - 1)$ 
10:  else
11:     $MSSL = Buff\_size - 2$ 
12:  end if
13:  if ( $LSS == MSSL$ ) then
14:     $SS = LSS$ 
15:  else
16:     $LSS = LSS + 1$ 
17:     $SS = LSS$ 
18:  end if
19:  if ( $SS > 0$ ) then
20:    Skip next sample
21:  else
22:    Go to 3
23:  end if
24: else
25:   Record new  $R$  in the buffer and forward it to the cluster head
26:   Go to 3
27: end if

```

cluster independently, we consider only one cluster comprised of five sensor nodes.

3.6.1 Evaluation of the temporal correlation-based adaptive sampling

3.6.1.1 Assumption

For the first part of evaluation, we use the presented data correlation model in Section 3.4.4 to analyse our adaptive sampling approach. As indicated in

Table 3.2: Sensor node specification

Unit	Type	Specifications	other
Microcontroller	Atmel Atmega128L	128 KB program memory	4KB RAM
Memory	AT45DB041B	512 KB	-
Radio	CC1000	speed: 38.4 Kbps	modulation: FSK
Power	Lithium	Size: 3B45	-
Sensor	SHT1X	Temperature	-

Section 3.4.4.1, there are three different versions of $MSSL$ adjustment that we evaluate to see which performs best.

In addition, we can also use different way to adjust the Last Skip Sample (LSS). In proposed solution, at each sampling point and for each accurate prediction, LSS increases by one until it reaches $MSSL$. This parameter can be removed from the algorithm, meaning that for every accurate prediction, SS has been assigned by $MSSL$. Therefore, the sensor node directly skips $MSSL$ number of samples for each accurate prediction.

Having alternative solutions to set $MSSL$ and LSS parameters means that there are different ways to adjust sampling frequency. Accordingly, we identify 12 different versions of the sampling frequency adjustment algorithm presented in Table 3.3. In this Table, the second column shows one of the $MSSL$ adjustment method presented in Section 3.4.4.1, the third column shows different possibilities to adjust LSS and the last column states whether sensor nodes predict skipped samples.

We evaluate all these possibilities in next section to show why the solution proposed in Section 3.4.4 is the best choice. In addition, we compare these scenarios with the solution suggested in [87].

Table 3.3: Possible scenarios of sampling frequency adjustment algorithm

	MSSL	SS	Prediction
Scenario 1	$\begin{cases} \text{if } d \leq CRe & MSSL = Buff_size - 2 \\ \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2) \times (1 - d) \\ \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\ \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = LSS$	-
Scenario 2	$\begin{cases} \text{if } d \leq CRe & MSSL = Buff_size - 2 \\ \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2)/2 \\ \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\ \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2)/2 \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = MSSL$	-
Scenario 3	$\begin{cases} \text{if } d \leq CRe & MSSL = Buff_size - 2 \\ \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2)/2 \\ \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\ \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2)/2 \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = LSS$	-
Scenario 4	$\begin{cases} \text{if } d \leq CRe & MSSL = Buff_size - 2 \\ \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2) \times (1 - d) \\ \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\ \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = MSSL$	-
Scenario 5	$\begin{cases} \text{if } d \leq 1 & MSSL = (Buff_size - 2) \times ((1 - d)) \\ \text{if } 1 < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = MSSL$	-
Scenario 6	$\begin{cases} \text{if } d \leq CRe & MSSL = Buff_size - 2 \\ \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2)/2 \\ \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\ \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2)/2 \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = MSSL$	-
Scenario 7	$\begin{cases} \text{if } d \leq CRe & MSSL = Buff_size - 2 \\ \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2) \times (1 - d) \\ \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\ \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = LSS$	✓
Scenario 8	$\begin{cases} \text{if } d \leq 1 & MSSL = (Buff_size - 2) \times ((1 - d)) \\ \text{if } 1 < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = LSS$	✓
Scenario 9	$\begin{cases} \text{if } d \leq CRe & MSSL = Buff_size - 2 \\ \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2)/2 \\ \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\ \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2)/2 \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = LSS$	✓
Scenario 10	$\begin{cases} \text{if } d \leq CRe & MSSL = Buff_size - 2 \\ \text{if } CRe < d \leq FRe & MSSL = (Buff_size - 2) \times (1 - d) \\ \text{if } FRe < d \leq (2 - FRe) & MSSL = 1 \\ \text{if } (2 - FRe) < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = MSSL$	✓
Scenario 11	$\begin{cases} \text{if } d \leq 1 & MSSL = (Buff_size - 2) \times ((1 - d)) \\ \text{if } 1 < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = MSSL$	✓
Scenario 12 (Improved version)	$\begin{cases} \text{if } d \leq 1 & MSSL = (Buff_size - 2) \times ((1 - d)) \\ \text{if } 1 < d \leq 2 & MSSL = (Buff_size - 2) \times (d - 1) \\ \text{if } d > 2 & MSSL = Buff_size - 2 \end{cases}$	$SS = MSSL$	✓

3.6.1.2 Data sampling

Figure 3.9 illustrates real data and sensor node predictions using the ARMA model represented in 3.4.2. As we see, sometimes readings are in range and sometimes not. For the time intervals in which data is close to the boundaries or fluctuates, the sensor nodes must sample more to be able to detect data that is out of range. For the times when data is close to the mean of the range or farther than the boundaries, more samples are skipped.

Figure 3.10 illustrates how frequently data is sampled. As seen in the figure, more samples are taken when data is closer to the boundaries and fewer samples are taken otherwise. Figure 3.11 illustrates the prediction error per sample point.

3.6.1.3 Accuracy

Figure 3.12 shows that for all scenarios, we are able to achieve at least 90% event detection accuracy (as requested by the user). However, those scenarios in which SS is updated based on LSS , show better accuracy levels. Scenarios such as 1, 3, 7 and 9 utilize this policy and for each accurate prediction increase their LSS by 1 until they reach the $MSSL$. By doing so, they can detect events early, and most importantly their average error is less than others (see Figure 3.13). In scenarios 2, 4 and 11, in which SS is updated based on $MSSL$, there is

Figure 3.9: Real data and predicted data

Figure 3.10: Sampling instances

Figure 3.11: Prediction error

likely to be a large number of skips. These skips prevent them from detecting events during these time periods. By adjusting $MSSL$ values, scenario 9 and our approach, which employ the first policy, yield the best accuracy level and average error. It means that at each sample point, $MSSL$ is adjusted based on its exact distance to the mean of the data, while other scenarios partially or completely use fixed values for $MSSL$ in different regions. Scenario 9 has a better result rather than our approach because, for each skipped sample, sensor nodes predict value for that epoch, while in case of our algorithm, the sensor

Figure 3.12: Event detection accuracy for different scenarios

Figure 3.13: Average error for different scenarios

node has no prediction for that epoch. As seen in Figure 3.13, the algorithm of [87] leads to a lower average error than our approach since it tracks every small change in sensor readings.

Figure 3.14: Average energy consumption for different scenarios

3.6.1.4 Energy consumption

Having a higher event detection accuracy and a lower average error results in additional energy consumption, as indicated in Figure 3.14. Scenarios which use the first policy ($SS = LSS$) take more samples to provide higher accuracy levels. The sensor nodes update their number of skips incrementally until $MSSL$ is reached. Therefore, they expend much more energy than other scenarios, which employ the second policy to update their SS . For the second policy, sensor nodes immediately skip $MSSL$ number of samples, which lessens energy consumption. Our algorithm and other scenarios expend less energy than [87]. Meanwhile, compared to most of the other scenarios, the energy usage of our algorithm is high, since it performs more calculation and is sensitive to any small difference between its reading and the mean of the data.

3.6.1.5 Energy-quality trade-off

Considering these results, an evaluation metric is required to compare energy and quality in term of error parameters together and find a trade-off between them. To do so, we use the *energy * error* metric. As shown in Figure 3.15, most scenarios improve or have the same performance as the approach of [87]. Our policy, labelled as the improved version, performs best. Therefore, this solution can be used as an improved mechanism which expends less energy while satisfying user-defined data quality.

Figure 3.15: Trade-off between energy and error

3.6.2 Evaluation of spatial-temporal correlation-based adaptive sampling

3.6.2.1 Assumption

In this section, we compare proposed spatial-temporal correlation approach with the modified version of two other existing techniques. The first one is an adaptive sampling approach (ASAP) proposed in [129] which in some ways is similar to our scheme. In ASAP, the cluster head is responsible for selecting sampler nodes that perform periodic sampling and send their readings to the cluster head. A non-sampler node samples only during the forced sampling periods. Sampling frequency is fixed and sampler nodes track small changes in sensor node readings.

As a second technique, We combine ASAP with the temporal correlation-based adaptive sampling proposed in [87] to develop a hybrid. In the hybrid approach, sampler nodes adjust their sampling frequency according to the technique proposed in [87]. We again use the dataset of [152], with temperature readings collected over two months. We assume that the data range is [21, 24], buffer size is 20 and data sampling interval is 2 h. The network architecture consists of several single-hop clusters. Since our spatial-temporal correlation-based solution is implemented locally inside each cluster, we consider only one cluster consisting of 25 sensor nodes to evaluate the solution.

3.6.2.2 Sampling

In this section we analyze how the proposed spatial-temporal correlation based sampling technique reacts to the changes in sensor node readings. For the time intervals in which the data is close to the boundaries or fluctuates, the sensor nodes need to sample more often in order to detect data that is out of range. When the data is close to the mean of the range or farther than the boundaries, samples are skipped more often. Following we represent two sampling scenarios for the readings which fall inside and around data boundaries, respectively.

Readings inside data boundary: Figure 3.16 illustrates readings of a sensor node whose readings are inside the boundaries and it experiences stability in environmental condition. Figure 3.17 illustrates how frequently data is sampled when data falls within the region boundaries. Figure 3.18 illustrates the prediction error for this data.

Figure 3.16: A comparison between the real data and the predicted data for reading inside the boundary

Figure 3.17: Sampling frequency for reading inside the boundary

Figure 3.18: Prediction error for reading inside the boundary

Readings around data boundary: Figure 3.19 illustrates the output of a sensor node whose readings are close to the boundaries and that experiences environmental instability. Sampling frequencies and event detection accuracy are illustrated in Figures 3.20 and 3.21. As the figures show, more samples are taken when data is closer to the boundaries and fewer samples are taken in all other cases.

Figure 3.19: Real data and prediction data for readings around the boundary

Figure 3.20: Sampling points for readings around the boundary

Figure 3.21: Prediction error for readings around the boundary

3.6.2.3 Transmission cost

We define the transmission cost in terms of the total number of packets transmitted in the network. Our approach has two kinds of packets: control packets and data packets. This approach also employs a distributed sampler node selection mechanism to select sampler nodes. As seen in Figure 3.22, this implies transmitting more control messages than the other two centralised approaches.

With regard to the data-transmission cost, our approach has the lowest number of messages, as Figure 3.23 demonstrates. Our approach reports a message only if it cannot accurately detect whether data falls in-bounds or out-of-bounds. In ASAP, all sensor nodes transfer data to the cluster head in the forced sampling times, while in the hybrid approach data is only transmitted when the prediction error increases. Therefore, ASAP has the highest transmission cost.

3.6.2.4 Data quality

Data quality is usually defined as the average error (the difference between real reading and transmitted data at the cluster head) in each cluster. Using this definition, as illustrated in Figure 3.24, our approach produces lower data

Figure 3.22: Total number of transmitted control messages

Figure 3.23: Total number of transmitted data packets

quality than the other two approaches. However, our approach is not designed to report any small changes and is based on in-bounds and out-of-bounds principles. As such, it leads to the lowest error when most of the readings are

near the boundaries or the readings are stable. Detecting whether data is in bounds or out is the main challenge for our approach. Figure 3.25 shows that our approach achieves at least 90% event detection accuracy (as requested by the user).

Figure 3.24: Average error

Figure 3.25: Event detection accuracy

Figure 3.26: Energy consumption

3.6.2.5 Energy consumption

Having a higher event detection accuracy and a lower average error results in additional demand for energy, as illustrated in Figure 3.26. Since ASAP and hybrid are sensitive to any small changes in the environment, they require more samples and more sampler nodes to monitor the area, which creates greater demand for energy.

3.6.2.6 Trade-off between energy consumption and data quality

Considering the results presented above, an evaluation metric is required to compare energy usage and data-quality metrics together and find the optimal trade-off between them. To do so, we again use the *energy * error* metric. As Figure 3.27 shows, our approach provides the best trade-off between error and energy parameters. This trade-off means that the higher error rates are compensated by lower energy consumption, which results in the lowest value for *energy * error*.

3.7 Chapter summery

This chapter has proposed an energy-efficient decentralized spatial-temporal correlation-based adaptive sampling method for wireless sensor nodes. The main idea behind this approach is to use a carefully selected, dynamically

Figure 3.27: Energy-error trade-Off

changing subset of sensor nodes to sense and report data and to predict the values of non-sampler nodes that are in sleep mode. Further, each sampler node tunes its sampling frequency based on the environmental conditions. This approach can effectively be used to prolong the network lifetime while satisfying the data-quality metric. In this way, the user or application introduces a data range within which sensor node readings are expected to fall. The proposed solution consists of four main steps: (i) sub-clustering, (ii) data prediction model building, (iii) sampler node selection and (iv) adaptive sampling. The temporal correlation-based adaptive sampling mechanism used by the sampler nodes has a decentralized sampling policy with which each sampler node decides its own sampling rate. The performance evaluation using a given dataset indicates a significant reduction in the power requirements using the proposed approach, while the event detection accuracy of 90%, as requested by the user or application, was met.

The solution proposed here ignores any uncertainty that may be present in the sensor node observations. We did not consider this uncertainty as it was not a significant issue in our dataset. However, if we face a significant systematic error in sensor node reading, our solution will not be energy efficient anymore. In this case, sensor nodes require to sample more to degreade the impacts of uncertainty. Covering this uncertainty can be considered as a future work.

Reward and Punishment-based Cooperative Adaptive Sampling

4.1 Introduction

To the best of our knowledge, most of the existing adaptive sampling methods ignore cooperation among sensor nodes to determine a proper sampling frequency for the entire network. Instead, each node decides its own sampling frequency based on its limited observation, or else the base station decides the sampling frequency for the sensor nodes. While the latter approach may lead to higher demand for energy due to higher data transmission, the former may lead to a global data-quality loss in order to lower demand. To address these two aspects simultaneously, this chapter presents a cooperative adaptive sampling approach based on a reward and punishment concept. The idea behind this approach is that when a sensor node detects frequent environmental changes, it increases its sampling rate, while other sensor nodes in its vicinity decrease their sampling rates to maintain the overall data quality of the whole network and keep energy expenditure low. The main contributions of this chapter are as follows:

- We define three different states for sensor nodes based on behavioural changes in the environmental data.
- We define a utility function for each sensor node, which aims at finding a good balance between its data prediction error and the remaining energy.
- We propose a cooperative adaptive sampling mechanism based on the reward and punishment concept, in which each node can select a sampling rate from its available sampling rates, i.e. its action set.

- Using this reward and punishment mechanism, we motivate sensor nodes to cooperate with each other. When some sensor nodes in a neighborhood experience frequent changes in the environment, other nodes can lower their sampling rate to let the former increase their sampling rate and keep the overall network energy expenses low.

4.2 Preliminaries

Before explaining our adaptive sampling method, we first explain the relevant basic definitions and assumptions.

4.2.1 Network architecture

The network architecture considered here is the same as that described in Chapter 3.

4.2.2 Data behavioural states

We consider three states for the data behavioral pattern: (i) stable state, denoted as A ; (ii) transition state, denoted as B ; and (iii) unstable state, denoted as C . To determine the state of the data behavioral pattern indicating the environmental states, we define two thresholds, θ_1 and θ_2 , against which we compare a node's current observation and prediction values.

In the A state (stable state), the data behavioral pattern rarely changes. This stability implies that the prediction error remains low (i.e., lower than θ_1). In this case the sensor nodes may have a low sampling frequency. In the B state (transition state), the data behavioral pattern starts to present some abnormal behaviors and more frequent changes occur. In this case the prediction error is between θ_1 and θ_2 . In the C state, many changes are observed in the data behavioral pattern, which may indicate the occurrence of an event. In this case, the prediction error is higher than θ_2 and the sensor nodes must use higher sampling frequencies. Our design principle is to help nodes in the C states increase their sampling frequency while nodes in the A or B states decrease their sampling frequency. By doing so they will combine to keep the network-wide energy consumption low and the data quality high. Comparing this with the proposed solution in Chapter 3, we see some similarity between region definition and the states defined in this chapter. However, the states defined

here are not restricted to the particular ranges and instead change on the basis of data behavior.

To track how the level of error generated on the cluster head when the sampling frequency is reduced, we assume that nodes use a linear regression model for predicting future observations. The cluster head and the sensor nodes all use the same prediction model. Sensor nodes must know whether they need to send their readings to the cluster head when they change their sampling frequency. Therefore, we define the following rules:

- If applying the new sampling frequency changes sensor node's state from A to B or if it causes no state change, the sensor node does not need to send its readings to the cluster head.
- If applying the new sampling frequency changes sensor node's state from B to A , sensor node needs to send its new readings to the cluster head and update its prediction model.
- If the sensor node enters the C state, it must send its new readings to the cluster head.

4.3 Reward and punishment-based adaptive sampling

Our algorithm has the following three major steps: (i) initialization, (ii) finding the new sampling frequency and (iii) deciding whether to change the sampling frequency at the node level. Each of these steps is explained in more detail below.

4.3.1 Initialization

In this phase, we define a set of possible sampling rates, denoted as follows:

$$SRSet_i = \{SR_{min}, SR_1, \dots, SR_{max}\} \quad (4.1)$$

To gain insight into the actual behavioral changes of the data, at the initialization phase, we force each sensor node to sample based on its maximum sampling frequency for one full day. We take this dataset as the ground truth. At the end of the first day, using the regression model, each sensor node predicts its future observations for all frequencies contained in $SRSet$. Based on

these predictions, each node i calculates its prediction error, denoted by Err_i , using the following formula.

$$Err_i = Err(SR) - Err(SR_{max}) \quad (4.2)$$

where Err_i is the difference between node i 's measurement using the maximum sampling frequency and the current designated sampling frequency.

In addition to this error, we use the number of data behavioral changes occurring during each time period as an indication of the degree of dynamicity (data change). We utilize the following equation to calculate the degree of dynamicity (DYN) for each time period and each sampling frequency.

$$DYN_i = \frac{\text{Number of data changes observed by node } i}{\text{Number of readings during the sampling period}} \quad (4.3)$$

Based on these two formulas, we create the Environmental Change Table (ECT), as shown in Table 4.1. Each entity (i, j) in this table consists of two parameters, i.e., Err_i and DYN_i . To relate the data behavioral pattern with the prediction error and identify which state nodes are currently in, we create a new table called $SRDY$ with the degree of dynamicity (DYN) as its columns and possible sampling rates contained in $SRSet$ as its rows, as Table 4.2 reveals. Each cell in this table represents the error measurement corresponding to a certain degree of dynamicity and sampling rate.

One should note that it is possible to have similar degrees of dynamicity and different prediction errors for different time periods and a single sampling frequency. In this case, the mean value of the prediction errors is used as the average error corresponding to that specific degree of dynamicity. These calculations and table creations are performed once and offline, and therefore create no additional burden on the resources of the wireless sensor network. Once all $SRDY$ tables are created for all sensor nodes, each node sends its $SRDY$ table to the neighboring nodes.

Table 4.1: Environmental change table (ECT)

	TP_1	...	TP_N
SR_1	(Err_{11}, DYN_{11})	...	(Err_{1N}, DYN_{1N})
SR_2	(Err_{21}, DYN_{21})	...	(Err_{2N}, DYN_{2N})
...	...	...	...
SR_S	(Err_{S1}, DYN_{S1})	...	(Err_{SN}, DYN_{SN})

Table 4.2: Sampling rate degree of dynamicity (*STDY*)

	DYN_1	...	DYN_N
SR_1	Err_{11}	...	Err_{1N}
SR_2	Err_{21}	...	Err_{2N}
...	...	...	...
SR_S	Err_{S1}	...	Err_{SN}

During the initialization phase, the cluster head assigns each node a maximum energy (TEg) that is allowed to be used at each time period. If the maximum amount of energy for a specific time period is not used, it is carried over to the next time period. The cluster head also assigns one of the sensor nodes as the decision-making node. The decision node, called *DNode*, is responsible for calculating and suggesting a new sampling frequency for each node based on the procedure described in Section 4.3.2. This selection procedure may be performed in the cluster head as well, but this changes it to a possible single point of failure. To decentralised this selection procedure and balance the power drain among sensor nodes, the decision node role is assigned to different sensor nodes at different time intervals. Moreover, acting as a decision node role is another motivation parameter for sensor nodes to cooperate with the rest of the network to achieve energy-efficient data gathering. Selection of the decision-making node is always based on the sensor node's energy level. To calculate the remaining energy at the end of each time period, we use the following formula:

$$REg_i = E_{i_{Last}} - (a \times E_{Sens} + b \times E_{Trans} + E_{Process}) \quad (4.4)$$

, where E_{Last} is the sensor node's residual energy at the end of the last sampling period, E_{Sens} is the amount of energy required to sense one bit, E_{Trans} is the amount of energy required for transmitting one bit, $E_{Process}$ is the amount of energy required for processing one bit, and a, b are the amount of sensed and transferred data.

To motivate sensor nodes to cooperate with each other to determine their sampling frequency, we define two parameters, namely the cooperation parameter (*CP*) and prediction parameter (*PP*). The former represents the node's cooperation level, while the latter indicates the node's prediction accuracy. If a sensor node cooperates with other nodes and accepts the suggested sampling frequency of the decision node, its *CP* parameter is increased at the beginning of each time period. Similarly, if the degree of dynamicity of the

new prediction of a sensor node is in line with its current degree of dynamicity, its PP parameter increases. We use the CP and PP parameters to reward or punish the sensor node. For each sensor node i , we compare cooperation and prediction parameters against two thresholds, TH_{CP} and TH_{PP} :

- $CP_i > TH_{CP}$ and $PP_i > TH_{PP}$: The sensor node cooperates with the network, and its predictions about the degree of dynamicity are correct. We consider this to be a trusted node.
- $CP_i > TH_{CP}$ and $PP_i < TH_{PP}$: The sensor node cooperates with the network, but most of the time its predictions about the degree of dynamicity are wrong. We consider this to be a normal node.
- $CP_i < TH_{CP}$: The sensor node does not cooperate, and we consider this node selfish.

Algorithm 4.1 presents the pseudo-code of the initialization step.

4.3.2 Finding new sampling frequency

The decision node receives the current sampling frequency, prediction error, residual energy, cooperation and prediction parameters of each node and uses the multiple choice knapsack problem (MCKP) optimisation technique[153] to find a new suitable sampling frequency for each node.

Algorithm 4.1 Initialization step

- 1: L_1 = Define sampling rate set $SRSet_i = \{SR_{min}, SR_1, \dots, SR_{max}\}$;
 - 2: Sample (for one day) with SR_{max} and complete data set for different time periods
 - 3: Predict observations for different sampling rates
 - 4: Find prediction error for each SR_{ij} in each TP_j : $Err_{ij} = Err(SR_i, TP_j) = Err(SR_i) - Err(SR_{max})$
 - 5: Find dynamicity parameter DYN for each SR_{ij} in each TP_j : $DYN_{ij} = DYN(SR_i, TP_j) = \frac{\text{Number of data changes}}{\text{Number of readings}}$
 - 6: Make ECT table
 - 7: Make $SRDY$ table
 - 8: Send $SRDY$ table to other sensor nodes
 - 9: Assign max available energy for every sensor node in each time period
 - 10: Set CP and PP parameters
-

The optimization algorithm when finding the best sampling frequency for each node must meet the following conditions:

- The amount of prediction error must remain lower than the predefined threshold θ_2 . This means that sampling frequencies causing errors higher than θ_2 are ignored.

$$Err_i \leq \theta_2 \quad \forall i \in [1...N] \quad (4.5)$$

- For each node, the required energy (REg_i^j) for sensing and transmitting data for the next time period must be lower than the maximum available energy for each time period (TEg_i^j) plus the current residual energy (REg_i^{j-1}).

$$REg_i^j \leq TEg_i^j + REg_i^{j-1} \quad \forall i, j \in [1...N] \quad (4.6)$$

In fact, the optimization algorithm minimizes the level of data prediction error and energy exhaustion for each time period, i.e.,

$$\min \sum_{i=1}^N \frac{Err_i}{REg_i} \quad (4.7)$$

The decision node runs the optimization algorithm with the last sampling frequency and prediction error of each node as its input and finds the degree of dynamicity for each node. It then finds the best sampling frequency for different nodes using the *SRDY* table and finds the degree of dynamicity of nodes and the calculated residual energy (Equation 4.4).

After receiving the new sampling frequency, prediction error and residual energy from the decision node, each sensor node decides whether to cooperate with the other nodes and use the suggested sampling frequency, or act selfishly and use its own predicted sampling frequency. If it uses the suggested sampling frequency, we reward it by increasing its *CP* parameters. Otherwise, its *CP* parameter is decreased.

To predict its own sampling frequency, the sensor node stores three recent readings and the degree of dynamicity parameters. At the beginning of each time period, it employs the regression model and its *SRDY* table to predict its new *DYN* for the current time period and its current sampling frequency. At the end of time period, it compares its prediction with its current *DYN*. If its prediction was true, its *PP* increases, otherwise it decreases.

Algorithm 4.2 shows the pseudo-code for finding a new sampling frequency.

Algorithm 4.2 Finding new sampling frequency

```

1: FOR each node
2:   Receive  $SR_i, REg_i, K_i, SR_{sugg}$ 
3:   IF ( $H_i == 1$ )
4:     { Consider  $SR_{sugg}$  as a fixed  $SR$  in KCMP
5:       Send  $SR_{sugg}$  to KCMP as a fixed  $SR$  }
6:   ELSE Send ( $SSR_i, Err_i$ ) to the KCMP as an input
7: Run KCMP and get deifferent new ( $SSR_i, Err_i$ ) for different sensor nodes
8: FOR each node
9:   IF ( $K == 1$ )
10:    {IF ( $TStamp == 0$ )
11:       $TStamp = X \times TStamp$ 
12:    ELSE {
13:       $TStamp = TStamp - 1$ 
14:      IF ( $TStamp == 0$ )
15:         $j = 0$ }}
16: Find new energy level
17: Find a node with max available energy level and with parameter  $j = 0$ ,
   Select it as next DNode
18: Send New ( $SR_i, Err_i, REg_i, K_i, DNode$ )

```

4.3.2.1 Changing sampling frequency

The decision on whether to use the suggested sampling frequency or use its own predicted frequency is made by each sensor node based on the following rules:

- Trusted nodes use their own predicted sampling frequency.
- Normal nodes are given the opportunity to choose between their prediction and the decision node's suggestion. If they cooperate and select the decision node's suggested sampling frequency, their CP parameter increases, or vice versa.
- Selfish nodes are not allowed to act as decision nodes for a certain time period. This is part of the punishment procedure. Furthermore, during this time period, selfish nodes are required to use decision node's suggested sampling frequency and cannot update their CP parameters. This means that during the punishment period selfish nodes must cooperate with the network, but will receive no reward for their cooperation.

During each time period, if the prediction error increases and consequently changes the state of the sensor node to the C state, we apply the following rules:

- Trusted nodes continue predicting their new sampling rates and use them. They increase their PP parameter if their prediction is in line with the current degree of dynamicity; otherwise they decrease it.
- Non-trusted nodes use the maximum available sampling frequency of the SRSet matching their current residual energy level.

At the end of each time period, each sensor node applies the following rules:

- If the previous and current degrees of dynamicity match, then the sensor node increases its PP parameter; otherwise it decreases it.
- If $CP > TH_{CP}$ and $PP > TH_{PP}$ then the sensor node is identified as a trusted node and uses the regression model and previous reading to predict its sampling frequency.
- It uses Equation 4.4 to update its residual energy.
- It sends its current sampling frequency, prediction error, residual energy and predicted sampling frequency to the decision node.

Algorithm 4.3 presents the pseudo-code for the decision regarding acceptance of the new sampling frequency.

4.4 Performance evaluation

In this section, we evaluate our method and present the trade-off between energy usage and data quality achieved by our technique. We compare our technique with the decentralized adaptive sampling technique presented in [136] and a hybrid method which is a combination of our method and the solution proposed in [142].

We use the same dataset [151] of the previous chapter and consider a network consisting of 20 sensor nodes measuring temperature. We do not consider spatial correlation, and the network topology is a star. We use the raw data captured in the dataset as the ground truth, i.e. data obtained with the highest sampling rate. We assume $\theta_1 = 1$ and θ_2 varies between 1.5 and 4. Our simulation analysis shows that this range can best represent the effect of the threshold. We repeat the simulations 20 times and present the mean results. Table 4.3 presents our simulation parameters.

Algorithm 4.3 Changing sampling frequency

```

1: Receive ( $SR_i, Err_i, REg_i, K_i, DNode$ )
2: Predict dynamicity ( $DYN$ ) for current time period
3: IF ( $CP > TH_{CP}$  and  $PP > TH_{PP}$ ) {
4:   Use  $SR_N$ 
5:    $H = 1$  }
6: ELSE IF ( $CP > TH_{CP}$ ) {
7:   IF (it selects  $SR_N$ )
8:      $CP = CP - 1$ 
9:   ELSE IF (it selects  $SR_{DN}$ )
10:     $CP = CP + 1$  }
11: ELSE IF ( $CP < TH_{CP}$ ) {
12:   IF ( $K == 1$ )
13:     Use  $SR_{DN}$ 
14:   ELSE {
15:     Use  $SR_{DN}$ 
16:      $CP = CP + 1$  } }
    During Time Period if error increases and put sensor node in state C
17: IF ( $CP > TH_{CP}$  and  $PP > TH_{PP}$ ) {
18:   Use  $SR_N$ 
19:    $PP = PP - 1$  }
    At the end of Time Period
20: IF ( $DYN_{curr} == DYN_{pre}$ )
21:    $PP = PP + 1$ 
22: ELSE
23:    $PP = PP + 1$ 
24: Update Energy Model
25: IF ( $CP > TH_{CP}$  and  $PP > TH_{PP}$ )
26:    $H_i = 1$ 
27: ELSE
28:    $H_i = 0$ 
29: Use regression model and previous readings to find  $SR_{sugg}$ 
30: Send ( $SR_i, Err_i, REg_i, H_i, K_i$ ) to DNode

```

Table 4.3: Simulation parameters

$SRSet$	$\{5, 10, 15, 30, 60\}$
θ_1	1
θ_2	$[1.5, 4]$
$\sum_{i \in n} Err_i$	≤ 0.15

4.4.1 Prediction error

Figure 4.1 shows the prediction error for one day. In our cooperative technique, prediction error increases as the node detects data behavioral change in the B state. Therefore, there is a significant difference in the error between the decentralized approach [136] and the hybrid method when the readings are in the B state. However, the error generated by our approach remains within the tolerable error of [136]. In the A and C states, all techniques generate similar prediction errors with minor differences.

4.4.2 Number of samples

Figure 4.2 shows the number of observations during one day. As it can be seen, in our cooperative technique, the number of samples increases as node detects data behavioral changes in the B state. Since the decentralized method can

Figure 4.1: Prediction error

Figure 4.2: Number of samples for one day

tolerate low errors, especially in B state, this approach employs higher sampling frequency. In the hybrid approach, different states are defined for sensor nodes, but the sensor nodes are forced to keep their error lower than a specified threshold, which comes at the cost of more sampling and transmission.

4.4.3 Effect of threshold

Employing θ_2 , our approach can tolerate more error and curbs the number of observations. Figures 4.3 and 4.4 show the effect of threshold θ_2 (i.e., the upper bound of the tolerable prediction error) on the prediction error and energy consumption. As the figure indicates, compared with [136] and the hybrid method, our technique conserves more energy while maintaining a similar prediction error for lower values of θ_2 . Smaller values of θ_2 lead to more data transmission and consequently more energy consumption. For larger values of this threshold, the sensor nodes can tolerate more changes and do not need to send data frequently. However, because they do not have this threshold and do not tolerate prediction errors, the decentralized method [136] and hybrid method [142] expend more energy.

Figure 4.3: Prediction error versus error threshold θ_2 Figure 4.4: Energy consumption versus error threshold θ_2

Figure 4.5: Trade-off between energy and error parameters

4.4.4 Trade-off between energy and error

To evaluate the trade-off between energy and error parameters, we again use the *Energy * Error* metric. As shown in Figure 4.5, when lower values are assigned for θ_2 , all approaches exhibit similar behavior. When increasing θ_2 , our approach provides a better trade-off between energy and error parameters. Thus, our approach can best satisfy increasing θ_2 .

4.5 Chapter summery

In this chapter, we proposed a cooperative adaptive sampling technique to satisfy both network lifetime and data-quality requirements. Instead of changing the sampling frequency based only on the energy level and data quality of a single node, we allow the sensor nodes to decide on changing their sampling frequency based on the network-wide energy level and global data behavioral pattern change. We encourage the sensor nodes to cooperate with each other to achieve the best trade-off between these two quality-of-service parameters by using the reward and punishment concept. Our technique outperforms a similar decentralized approach in terms of energy consumption while maintaining a similar prediction error. Performance evaluation and comparison with an existing decentralized adaptive sampling approach and hybrid method using a given dataset indicates that our approach saves more energy for the times

when sensor nodes fall into transient state, while for the other states the difference is not as great.

This approach dose not consider the spatial-temporal correlation among sensor nodes which can lead to more efficient sampling adjustment and sensor nodes scheduling. Therefore, building a cooperative sampling model based on the inter and intra correlation among sensor nodes can be considered as a future work.

Part II

Compressive Sensing

Figure 4.6: Organization of the thesis

Data Compression: State of the Art

5.1 Introduction

Data compression is one of the well-known data reduction techniques applied in sensor networks to cope with resource restriction challenge in these networks. Proposed compression techniques can be considered in three different levels: compression in sensing level which reduces the number of samples, compression in computational level which reduces the number of bits and compression in data packets which reduces the number of data packets [154, 90].

The compression at sensing level has been already investigated in the first part of this thesis. The second part of the thesis studies data compression at computational and communication levels. Data compression is a process in which the amount of data or transmission time, or both, is reduced. Due to the limited processing and storage resources of the sensor nodes, data compression algorithms for wireless sensor networks aim to find an efficient way to compress data to reduce the energy costs of individual sensor nodes and improve the synthesized ability of the entire network [89, 88].

This chapter classifies and reviews existing data compression techniques and highlights the use of compressive sensing for wireless sensor networks. Existing data compression techniques can be broadly divided into two categories: (i) lossless compression and (ii) lossy compression. Each of these can be further sub-divided into local and distributed approaches. Each of these categories is described in more detail below, after which we offer a comparative overview.

Figure 5.1: Lossless data compression classification

5.2 Lossless data compression

Lossless data compression is an appropriate choice for applications requiring high precision. It is obvious that high compression ratio and high precision are often in conflict. When we need certainty that the exact same data will be obtained after decompression, use of lossless compression methods is the only option.

Lossless data compression reduces the size of data in such a way that the decompression function can restore the original data exactly as it was. The basic assumption of lossless compression techniques is that the data contains redundant information. Therefore, it can be condensed using statistical modelling techniques that use the probability of a character or phrase appearing in the data. These statistical models can then be applied to generate codes for occurrence of specific characters or phrases based on the probability of their occurrence and assign the shortest codes to the most common data [93, 89].

Lossless compression techniques suffer from one major limitation which makes them less suitable for wireless sensor networks: they do not take advantage of the temporal correlation that occurs in many wireless sensor network applications. Exploiting the redundancy among samples from a single sensor node can lead to reduction in the volume of data to be transmitted in wireless sensor network [90].

Figure 5.1 shows classification of the lossless compression techniques.

5.2.1 Local lossless data compression

Local lossless compression can be divided into three categories: dictionary-based, entropy-based, and run-length-based compression techniques.

5.2.1.1 Dictionary-based data compression

The Lempel-Ziv-Welch (LZW) algorithm [155] is the reference dictionary-based compression algorithm utilized and extended in the sensor network area. Dictionary-based compression solutions proposed in [156] emphasize finding patterns that frequently occur in sequences of data. This approach builds a dictionary step by step. To do so, it fills indexes in the dictionary with single characters, then searches the patterns of input data to find a pattern that is not in the dictionary, and finally adds that pattern to the dictionary. The index of this pattern is the code word, which will be replaced in the data streams. This procedure repeats and adds a new index for any new patterns occurring in the data stream. The proposed solution is simple and energy efficient, which removes data-transmission overhead.

Packet loss is a parameter common in wireless sensor networks, while it is not properly addressed in LZW. The authors in [125] develop a dictionary-based algorithm to cover this issue. Sensor-LZW (SLZW) employs the temporal correlation among each sensor node reading and divides each sensor node stream into small blocks. Inside each block, the LZW algorithm compresses the stream inside the block into several packets. In this way, sensor packets inside each block are independent of other blocks while depending on the packets inside their blocks. This relationship means that the loss for each packet influences only the other packets located in the same block, while other blocks are isolated. Furthermore, SLZW utilizes the dynamicity of the input data stream and frequency of the pattern occurrence to dynamically change the index code length. The memory spent to store dictionary data satisfies the resource limitations of the sensor nodes; however, the size of the dictionary can grow, which affects compression efficiency.

To address this issue, Yan-Li et al. [91] present a compression algorithm which seeks the appropriate dictionary size to avoid the dictionary growth issue. Despite the LZW method, this technique ignores the initialization of the dictionary with single characters and applies restrictions on the pattern size inside the dictionary. As a result, this approach improves memory usage and compression efficiency. In addition, it compares the patterns observed in two

adjacent sequences of data to find the incremental part and minimizes the size of the compressed data.

A dictionary-based compression algorithm is proposed in [92] that proposes compressing provenance information in sensor networks. To build a dictionary, each sensor node ID is considered as an index in this dictionary, and a route traversed by the sensor node packets represents a stream of indexes in the dictionary. There is a high chance that the routes to transfer data packets can be partially or entirely the same for different transformations. This statement is employed to find the common sub-streams of sensor node indexes between past and current data-transmission routes. As a result, these common indexes can be mobilised to compress the transformation routes of all future transmission routes. Minimizing the size of the compressed data-transmission path leads to smaller data packet sizes and less data-transmission cost. However, the proposed solution is dependent on topology. In the case of dynamic topology changes, the common data-transmission routes change more frequently; therefore no dictionary of sensor node indexes can be constructed.

5.2.1.2 Entropy-based data compression

Entropy-based encoding is a coding scheme which represents symbols with a minimum number of bits. Huffman coding and arithmetic coding are the most common entropy-based encoding techniques [157].

lossless entropy compression (LEC) is a temporal correlation-based compression technique proposed in [158] that focusses on differences between consecutive sensor readings. These differences are divided in alphanumeric characters into different groups. Inspired by Golomb coding [66] and Elias coding [159], all compressed data is represented by concatenating fixed and variable-length codes where the group and the group index are highlighted by variable and fixed codes, respectively. This approach ensures an effective compression ratio, but the algorithm is static and hence cannot adapt to the changes in the source data statistics.

An adaptive Huffman algorithm introduced in [160] employs temporal correlation to propose an adaptive compression solution. Minimizing the number of levels in the Huffman binary tree is the main contribution of this approach, and this increases compression efficiency at the cost of intensive computation.

The authors of [161] devise a predictive coding-based data compression technique. In this approach, the base station builds a data prediction model for each sensor node and sends the copy of that model to the sensor nodes that will later support data prediction. Each sensor node predicts data and

compares the predicted data with the observed data. The error between the predicted and original data is encoded and transferred to the base station. The base station decodes this error and adds it to the predicted data produced by the identical prediction model. The drawback of this compression algorithm is that it is computationally intensive.

The adaptive lossless data compression (ALDC) algorithm proposed in [162] is an extended version of the LEC method. This algorithm supports compression of different types of signals with satisfactory compression ratios. To do so, it employs three different code tables to find the best-compressed value for each data type. Each sensor node collects a sequence of data samples, calculates the difference between them and then compresses data using each of these three code tables. The compressed data with the smallest size is selected as the compressed version and transferred to the base station. This algorithm is suitable for both real-time and delay-tolerant transmission and improves compression efficiency. However, each sensor node must store three different code tables and compress and test data with each of these tables, which results in more computational and storage overhead.

5.2.1.3 Run-length based data compression

One of the simplest and most popular data compression algorithms is run-length encoding (RLE) [163]. This approach focusses on a data stream or files that contain several occurrences of the same sub-streams or patterns inside data streams and files. It then replaces each pattern or sub-stream with the number of its repetition. Run-length encoding is based on the same consecutive input stream, and its results depend on the data source.

K-RLE is an RLE method that improves the compression results even when the redundancy of the input sequence is not high [164]. In this approach, each sensor node checks whether data pattern d occurs in $d + K$ and $d - K$ for n consecutive times, then replaces that pattern with nd . The variable K represents the precision of the algorithm. If we increase the value of K , the compression ratio increases while the precision of the algorithm decreases. Therefore, selecting an appropriate value for K is the main challenge in this method.

5.2.2 Distributed lossless data compression

5.2.2.1 Distributed source coding based data compression

Distributed source coding (DSC) is one of the promising data compression solutions that has emerged for data and image compression. Distributed encoding of spatially correlated data from individual data sources and centralised

joint decoding are the main characteristics of this technique. This concept is well suited to the wireless sensor networks context [88].

In this technique, each sensor node encodes its data separately and forwards it to the base station. The base station is a joint decoding centre that receives separate encoded data from sensor nodes and then employs spatial correlation among the sensor node readings to jointly decode the data received. A simple encoding procedure in low-power sensor nodes and complex decoding process at the powerful base station correspond well to the sensor network resource restrictions. Several studies propose solutions to exploit the richness of information provided by spatial correlation of measurements taken from sensor nodes and further compress the measurements without inter-node communications, which is characteristic of distributed source coding [165, 166].

The energy-efficient adaptive distributed source coding (EEADSC) proposed in [167] is one of the first source-coding methods applied in wireless sensor networks. This technique is implemented in two steps: training and compression. In the training step, the algorithm finds the spatial correlation among sensor nodes. In addition, it groups sensor nodes into different clusters and then runs the compression step for each cluster. Since the compression is applied in only one cluster, it ignores the overall correlation among all sensor nodes in the network, which leads to lower compression efficiency. However, it improves energy efficiency by using local data compression.

Oldewurtel et al. [168] studied the performance of distributed source coding algorithms in terms of their power consumption for a number of network topologies in combination with cluster head selection methods. They analyze the energy efficiency of distributed source coding on grid, random point process and Thomas PP topologies, and for each topology classify sensor nodes inside clusters. Furthermore, they examine different cluster head selection effects on network performance. Their analysis results show that applying distributed source coding on Thomas PP topology increases the energy efficiency of the network better than the other two topologies.

One study [169] proposes a Slepian-Wolf-based compression method [170], which emphasizes combined optimizing of source coding rate and data routing to improve compression efficiency. Inspired by game theory, the coding rate allocation problem is modelled as a cooperative game. In this game, the sensor nodes cooperate with their neighbors to satisfy joint entropy requirements while each sensor node tries to lower its demand for power by lowering its coding rate. A range of available coding rates is defined and each sensor nodes selects an optimal source code rate from this range. On the other hand, this method links the coding rate allocation problem to the data-transmission

Table 5.1: Lossless data compression methods

	Data quality	Energy efficiency	Scalability	Compression efficiency	Type	Simplicity
[156]	-	✓	-	✓	Dictionary	✓
[125]	✓	-	-	-	Dictionary	✓
[91]	✓	-	✓	✓	Dictionary	-
[92]	✓	✓	✓	✓	Dictionary	-
[158]	-	✓	-	✓	Entropy	-
[160]	-	✓	-	✓	Entropy	-
[161]	✓	-	✓	✓	Entropy	✓
[162]	✓	✓	✓	✓	Entropy	-
[164]	✓	-	-	-	Run-Lenght	✓
[163]	✓	✓	✓	✓	Run-Lenght	-
[167]	✓	✓	-	-	Distributed Source Coding	✓
[168]	✓	-	-	✓	Distributed Source Coding	✓
[171]	✓	✓	✓	✓	Distributed Source Coding	-
[169]	✓	✓	✓	✓	Distributed Source Coding	✓

policy in order to reduce traffic congestion in the network. To do so, a dynamic data-transmission policy is proposed that defines and uses a congestion price parameter to assign a weight to the link between each sensor node and its correlated neighbor nodes. Each sensor node selects its optimal coding rate and its next relay node based on this congestion price, which can diminish network congestion problems. The outputs of this cooperative game are reduced data redundancy in the network and improved network lifetime.

A hierarchical distributed source-coding algorithm combined with an optimal data-transmission strategy is presented in [171]. Two-tier network architecture is utilized to implement a hierarchical distributed source coding algorithm. The base station makes use of this architecture as a data-transmission route and gathers sensor nodes' raw data. It then builds the coding topology based on the correlation coefficient and sends the relevant coding topology data to the sensor nodes in order to encode their readings. On the other hand, due to the large scale of sensor networks, coded data packets traverse a long path from the sensor nodes to the base station. Therefore, they more likely experience a higher level of data packet loss. To cope with this issue, the approach proposes an optimal transmission-scheduling algorithm based on particle swarm optimization. This algorithm is implemented in the base station in two phases: in the offline phase, it draws on coding data to produce an optimal retransmission rate. In the online phase, a transmission information-based feedback system is then initialized that adjusts the retransmission rate for the coded packets based on their importance. This solution outperforms static source coding methods in terms of energy efficiency.

Table 5.1 summarizes the aforementioned approaches and their support for various quality of service parameters. We evaluated the performance of each

method based on the data quality, energy efficiency, scalability and latency parameters. Our studies revealed that lossless solutions provide better data quality which comes at the cost of more energy consumption. Dictionary and entropy based solutions represent a good compression efficiency, however, the existing spatial/temporal correlation in sensor networks are not utilized which limits their energy efficiency. On the other hand, distributed source coding based methods employ spatial/temporal correlation among sensor nodes and their readings and achieves higher compression and energy efficiency.

5.3 Lossy data compression

A lossy compression technique, as we have stated above, is a compression method in which some data is discarded during the compression process and the amount/size of compressed data is less than the original input [93, 89].

Figure 5.2 shows the classification of lossy compression techniques.

Figure 5.2: Classification of lossy data compression techniques

5.3.1 Local lossy data compression

Local lossy compression techniques can be divided into two categories: entropy-based and run-length-based compression techniques.

5.3.1.1 Entropy-based data compression

Differential pulse-code modulation (DPCM) [172] is widely used for compressing signals, especially for speech and video coding. It is based on the concept that efficient decoding of the difference between highly correlated neighboring samples will lead to high compression efficiency [173].

The authors of [174] propose an entropy-based compression solution for sensor networks. Encoding based on DPCM is applied among temporally correlated sensor node observations. Furthermore, predictive coding helps to encode spatially correlated observations that are exchanged among sensor nodes. Accurate decoding at the base station satisfies the accuracy requirement of the application. However, this method suffers from high data communication costs among sensor nodes, which results in higher energy consumption.

Similarly, an adaptive DPCM-based compression approach is introduced in [175], which utilizes temporal correlation among sensor node readings to compress data. This approach focusses on making a trade-off among information loss, energy usage and compression ratio. This trade-off is achieved by applying multi-objective evolutionary algorithms [176] to the sensor node readings, which can be adjusted to satisfy various application requirements. Each sensor node encodes its readings through this evolutionary algorithm and forwards it to the base station. To decode sensor node readings, the base station must know the location of each sensor node. The distinguishing features of this solution are accurate data reconstruction in a noisy environment and significant energy savings. In this representation, most data loss at the sensor node level is considered noise, which helps to decode sensor observations accurately.

A spatial-temporal correlation-based compression technique is presented in [94] to compress multiple data types produced by heterogeneous sensor nodes. The proposed solution applies DPCM-based encoding on temporally correlated observations and distributes source encoding on spatially correlated observations. To achieve accurate representation of correlations among different data types in the decoder, a Copula function-based statistical model is applied. Simplicity, flexibility and accuracy are the main advantages of this solution. However, the energy efficiency metric is not discussed in detail.

5.3.1.2 Run-length based data compression

The authors of [177] address a lightweight temporal compression (LTC) technique for climate-monitoring applications. For each sensor node, LTC considers short time frames to check sensor readings and tries to find a linear trend among those readings. The data accuracy requirement is defined as an error bound, which allows LTC to propose higher compression ratios for higher error bounds.

An induced redundancy-based lossy data compression algorithm (IR-LDCA) is proposed in [178], which employs data correlation among sensor node readings to compress data. Striking a balance between data loss and compression ratio is the main challenge for this solution. To do so, each sensor node encodes its observations such that it can satisfy user-defined minimum data accuracy requirements. Employing higher compression ratios results in a significant reduction in data-transmission costs.

5.3.2 Distributed lossy data compression

Distributed lossy compression techniques can be divided into three categories: distributed source modelling (DSM), distributed transform coding (DTC) and compressive sensing.

5.3.2.1 Distributed source modeling based data compression

These techniques aim to find a function or model that best fits the input data. These functions or models may be parametric or non-parametric. Parametric models consider the input data as a random process that must be optimally estimated given the statistical parameters, such as mean and variance. Non-parametric models require very little prior information about the nature of the input data and are considered to be very robust [179]. This category is discussed above in Chapter 2, so we do not elaborate on it in this chapter.

5.3.2.2 Distributed transform coding based data compression techniques

The transform coding techniques decompose the source output based on transform theories into components and coefficients that are then coded according to their individual characteristics. This compression approach is widely used, especially for image, video and audio compression. Energy compaction is the main characteristic of transform coding, which represents the greatest proportion of signal energy by a few coefficients of the signal. Transform coding exploits this characteristic to convert signal coefficients from one domain into a few important and large coefficients in another domain. These few transformed coefficients are the most informative coefficient of the signal [180].

Classic transform coding techniques in the sensor network area must be adapted to the resource restriction characteristic of sensor nodes. The Fourier transform (FT), cosine transform, wavelet transform (WT) and Karhunen-Loève transform (KLT) are the most popular transform coding methods [180]. Full implementation of these techniques in wireless sensor networks is difficult. The main reason for this difficulty is that they often require availability of full knowledge of the network in one of the sensor nodes to calculate the transform coefficients. This requirement unnecessarily increases the inter-node communication between nodes, and consequently the demand for energy. Recent literature on wireless sensor networks has proposed a distributed version of the classic transform-based compression techniques for this context [89, 90].

We elaborate on some of these transform coding methods below.

Karhunen-loève transform The Karhunen-Loeve transform (KLT) is the optimal transform basis for precise extraction of important signal coefficients. However, it demands higher computational resource requirements. The higher energy compaction in this transformation results in reflecting important signal properties in a few signal coefficients [180, 90].

Amar et al. [181] propose a distributed KLT-based compression technique to remove the data redundancy in the network. In this technique, the data vector produced by sensor nodes is divided into sub-vectors in which each sensor node applies part of the KLT on its sub-vector, produces encoded data and transfers it to the base station. The base station then reconstructs the original data vector. Furthermore, it employs a greedy algorithm to build encoding matrixes. In each step of this algorithm, the base station finds one of the encoding matrix rows and appends it to the matrix based on the mean square error parameter. This algorithm is simple, however, its power consumption parameter is not evaluated in the published study.

A distributed adaptive covariance matrix eigenvector estimation (DACMEE) algorithm is proposed in [182] to facilitate KLT-based compression in sensor networks. Unlike the other compression solutions, which build encoding matrixes at the base station, this approach relies on both in-network local processing on the sensor nodes and information exchange among neighboring nodes to decentralize the building of the encoding matrix. Each sensor node applies the encoded version of its readings to locally produce its eigenvector of the encoding matrix. Having the local version of the encoding matrix, the sensor nodes are not required to transfer their readings to the base station to produce a network-wide encoding matrix. Each sensor node transfers only a compressed version of its readings to the base station. Furthermore, each sensor node is equipped with the Q -dimensional projections of the full set of sensor signal observations, which let it estimate the KLT-based approximation of the network-wide observations. The proposed algorithm is only applicable in fully connected or tree-based topology networks. Due to its in-network processing and localized encoding procedure, the suggested approach minimizes the communication and computation costs, which results in reduced energy usage.

Wavelet transform: Lifting and graph-based wavelet transforms are often applied to sensor networks. The lifting WT consists of three main steps: the splitting step, prediction step and updating step. In the splitting step, the sensor node observations are categorized in odd and even sets. In the prediction step, even observation coefficients then filter odd observations, while coefficients of odd observations filter even observations in the updating step [180, 88].

A distributed lifting WT compression algorithm was suggested in [171] to achieve an energy-efficient data aggregation solution. Before compressing sensor node readings, the proposed solution employs spatial correlation among sensor nodes to build and organize sensor nodes into clusters. Grouping highly correlated sensor nodes within each cluster allows a few sensor nodes to be representative of other neighboring nodes, which decreases data-transmission volumes. The representative sensor nodes of each cluster send their readings to the cluster head. The cluster head receives those readings and stores them in a two-dimensional measurement matrix. It then applies distributed fast wavelet compression to this matrix, produces a compressed version of the sensor reading and forwards it to the base station. The base station uses a three-step lifting wavelet recovery algorithm to recover the original data. Utilizing spatial correlation to cluster sensor nodes and spatial-temporal correlation to compress

sensor node readings results in higher data accuracy in an energy-efficient solution.

Lifting-based WT was suggested in [183] for the sensor nodes deployed in a regular grid topology to produce 1D wavelet transformation. Furthermore, this study combines this transformation with a tree-based routing method to produce a 2D wavelet measurement matrix. This tree-based routing approach allows one to calculate and transform sensor node coefficients from leaf nodes towards the root. In other words, leaf sensor nodes calculate the partial coefficient of a signal and forward it to the next sensor node in the tree. That sensor node adds its coefficient to the coefficients that are received and forwards it to the next sensor nodes. This procedure repeats along the path until all coefficients are calculated and forwarded to the base station. The proposed solution restricts communication overhead among sensor nodes; however, most of the sensor network applications are deployed in an irregular network topology, which makes the proposed solution topology-dependent. To cope with this issue, the authors extend their solution for irregular deployment of sensor networks [184]. They utilize a minimum-spanning tree method to implement WT on sensor node readings.

The authors of [185] present a robust information-driven data compression architecture (RIDA) for a large-scale dense wireless sensor network. This approach is applicable for cluster-based networks which experience slow variation in sensor readings. Inside each cluster, a logical mapping approach is utilized to rearrange sensor nodes and assign a virtual index for each sensor node. This mapping aims to group more highly correlated sensor nodes together to achieve higher compression efficiency. The cluster head forwards this mapping information to the base station. Sensor nodes exchange data with each other and then apply distributed cosine transform or distributed WT compression algorithms to produce signal coefficients and transfer them to the base station for reconstruction. The mapping information in the base station helps achieve more accurate data reconstruction. A resiliency mechanism is also proposed to address packet loss in the network. Using mapping information other than the compression algorithm results in higher energy savings while maintaining acceptable data accuracy.

The lifting WT suffers from high communication overhead. It is more likely that sensor nodes will transfer data twice. The graph-based WT is proposed to cover these shortages [186]. This technique localizes the lifting procedure in graph nodes by transferring data only once in a unidirectional path towards the base station. To do so, data transmissions are performed along a unique arbitrary routing tree. Observations from odd sensor nodes in the graph exploit

the correlated data of their odd neighbor nodes, and the process is repeated for the even nodes. In this solution, the splitting step of the lifting transform happens over the graph, and the updating step happens locally on the neighbor nodes.

5.3.3 Compressive sensing

Compressive sensing is a concept from the field of signal processing. The promise of compressive sensing is that it can reconstruct sparse or compressible signals from a small number of measurements without having prior knowledge of the signal structure. Unlike the traditional compression techniques, compressive sensing based approaches sample signals with a sub-Shannon-Nyquist rate and recover them with high accuracy if certain conditions are satisfied. In fact, compressive sensing regards the information rate instead of the sampling rate to sample and recover the signal [95].

Based on the nature of correlation used by the compressive sensing techniques, they can be divided into three categories: temporal-based, spatial-based, and spatial-temporal-based compressive sensing.

1. **Temporal correlation based compressive sensing:** The study presented in [110] introduces a belief-propagation-based compressive sensing technique that considers the temporal correlation model of the signal as given information and employs it to reconstruct the signal. To model temporal correlation, one can decrease the number of required measurements and help to achieve better reconstruction accuracy with a specific spike-and-slab Markov model. The accuracy of this Markov model changes over time, and this can influence reconstruction accuracy. Therefore, this algorithm periodically analyzes the model parameters and employs an online estimator to predict its parameter. This estimator is based on sequential expectation maximization (EM) algorithm [187] that is derived by maximizing the Kullback-Leibler (KL) [188] information measure. At the end of each reconstruction period, this estimator refers to reconstructed signal coefficients for the last time period to update the Markov model parameters. This approach focusses on robust accurate data reconstruction and decreases the number of measurements. However, the energy efficiency of this solution is not addressed explicitly.

The authors of [189] discuss how to make the encoding process more efficient in terms of resource management strategy. They proposed a compressive sensing approach that considers temporal correlation among

sensor node readings to recover the original signal. They combined their compressive sensing approach with gossip-based routing to recover the sparse signal in multi-hop wireless sensor networks.

In the case of telemonitoring applications, most signals, like ECG signals, are not sparse in the time domain. Therefore, current compressive sensing solutions cannot provide accurate reconstruction. To address this problem, one study [190] addresses a block-sparse Bayesian learning-based compression technique. The proposed solution represents an extreme performance improvement in recovering non-sparse signals. Furthermore, using the block structure to recover signals lessens reconstruction complexity, which results in lower computational cost and better accuracy.

A combination of model-based data prediction and compressive sensing is addressed in [191] to achieve energy-efficient data gathering in sensor networks. Compressive sensing facilitates the compression of sensor node readings while an adaptive data prediction model is built on this basis, reducing the data sampling frequency. Using this data prediction model, sensor nodes do not require any prior information regarding monitored signals and can adapt their sampling rates to the time-varying sparsity level of the signals. Minimizing the data-sampling rate based on the sparsity level of the signal provided by compressive sensing attenuates the data-transmission rate, which conserves power.

2. **Spatial correlation based compressive sensing:** The authors of [192] introduce a compressive sensing approach that employs amplitude-modulated analogue transmissions to transfer data. Furthermore, the sensor node does not require prior knowledge of sensor data and targets to recover signals from spatially distributed sensor nodes. In this approach, the sensor nodes are distributed in a single-hop network and directly transfer their readings to the base station. Each sensor node utilizes its local projection vector to encode its readings. This approach reduces in-network communication overhead, but synchronization among sensor nodes remains the main challenge.

Luo et al. [193] present a compressive data-gathering (CDG) method, which combines data compression and routing for data collection in large-scale wireless sensor networks. The result of this combination is a balanced distribution of energy drain over the network, which improves network lifetime. Furthermore, it shows a significant reduction in com-

munication cost. With regard to scalability, this solution cannot be applied in small-scale networks and large-scale dynamic networks.

A further study [194] introduces a joint sparse signal recovery method which assumes sensor nodes have been distributed into different clusters, and uses a joint sparse signal recovery mechanism (instead of an independent reconstruction process) to recover the compressed data. This solution minimizes energy expense in terms of data compression and transmission rates. However, it does not ensure that application-defined data accuracy requirements are met.

The authors of [195] target the event-detection application of wireless sensor networks in which the number of sensor nodes detecting an event is considerably lower than the number of all sensor nodes in the network. This may represent sparsity of events. The proposed compressive sensing-based event detection approach recovers the signal through a Bayesian reconstruction algorithm. Significant reductions in the number of measurements, as well as increased event detection probability, are the main performance achievements of this solution.

A compressed network coding-based distributed data storage (CNCDS) method is proposed in [196]. It takes advantage of spatial correlation among sensor node measurements and aims primarily to achieve good recovery accuracy. Furthermore, it concentrates on reducing the number of data transmission and reception parameters, which results in energy-efficient data gathering. To do so, the proposed solution adjusts the probability of forwarding received data packets to the number of neighbor nodes. This approach ignores the noise and error parameters in the transmission scenario.

The research presented in [96] is one of the first studies to propose a compressive sensing-based data-gathering solution based on a random walk algorithm. The authors employ this algorithm to produce a non-uniform random measurement matrix. To do so, randomly selected sensor nodes in each walk generate a measurement based on linear combination of both the node itself and neighboring nodes' observations. At the end of each random walk, the measurements produced are forwarded to the base station. Compared to the existing random projection solutions, this approach shows a significant reduction in data-transmission cost.

3. **Spatial-temporal correlation based compressive sensing:** A compressive sensing-based false data detection and correlation framework for

crowd sourcing applications is presented in [197]. In crowd sourcing applications, a large number of participants employ their mobile devices to collect environmental data and report it to the base station. Misbehaving or erroneous participants can produce faulty data and packet loss in the network. Missing or faulty data is reconstructed with a spatial-temporal correlation-based compressive sensing technique. This approach considers collected data as multidimensional time-series and groups sensor nodes into smaller grids based on their location. In the first step, this solution gathers raw data from participants as a training data set and groups them into different grids by considering spatial correlation among sensor nodes. Then, the compressive sensing algorithm reconstructs the sensor data with spatial-temporal correlation and the low-rank feature among sensor node readings. The proposed solution guarantees high data quality in environments with high packet loss and noisy characteristics, but the study did not address energy savings.

A hybrid network coding and compressive sensing-based solution in [198] represents a clustered spatial-temporal correlation-based compressive sensing method. Each sensor node applies a well-defined Gaussian code matrix to encode its reading and forwards it to the base station. Proper selection of the measurement matrix and network coding coefficients is utilized at the base station to implement a low-complexity data reconstruction algorithm that reconstructs data more accurately. In addition, the proposed solution reduces the number of measurements, resulting in energy-efficient data transmission.

The authors of [199] proposes a novel compressive sensing-based data-gathering solution for massive lossy data-transmission scenarios. The proposed compressive sensing solution models environmental data dynamics using an environmental matrix (EM) and then employs the spatial-temporal correlation observed in this matrix to reconstruct the data. Furthermore, this approach has a multiple attribute assistant (MAA) component, which correlates environmental attributes to discover data loss patterns in reported encoded data. Combining MAA and EM, the compressive sensing-based reconstruction method provides better reconstruction accuracy. However, this approach suffers from higher computational complexity.

Another study [111] employs sequential data gathering and progressive data reconstruction to address a sliding window-based compressive sensing solution. Sensor nodes use this method to periodically encode

and deliver their readings to the base station. The progressive data reconstruction method at the base station exploits Kronecker sparsifying [180] to recover the portion of sensor node readings reported during last time period. This progressive method uses the recovered sensor readings from the last time period to restore encoded data in the current time period. There is a trade-off between data reconstruction accuracy and reconstruction complexity, which is adjusted by the sliding window size. In addition, the dynamic adjusting window size modifies the number of required measurements, which results in adaptive data transmission.

Table 5.2 shows a comparison of the aforementioned lossy compression approaches based on the various quality of service parameters. Our studies revealed that lossy data compression techniques show better compression ratios which results in better energy efficiency. The transform based compression and compressive sensing based techniques propose higher energy efficiency and data compression ratios. These solutions do not require any prior information about the sensor nodes readings and their correlation. However, they are not scalable for the large scale networks. Compressive sensing solutions represent simple encoding and complicated decoding procedures which best suits the

Table 5.2: Lossy data compression methods

	Data quality	Energy efficiency	Scalability	Compression efficiency	Type	Simplicity
[173]	✓	✓	-	✓	Entropy	-
[174]	✓	-	-	✓	Entropy	✓
[175]	✓	✓	✓	✓	Entropy	-
[94]	✓	-	✓	✓	Entropy	-
[177]	✓	-	-	✓	Run-Lenght	✓
[178]	-	✓	-	✓	Run-Lenght	-
[181]	✓	-	-	✓	DTC(KLT)	✓
[182]	✓	✓	✓	✓	Distributed Transform Coding(KLT)	-
[183]	✓	-	-	-	Distributed Transform Coding(WT)	✓
[184]	✓	✓	-	✓	Distributed Transform Coding(WT)	-
[186]	✓	✓	✓	-	Distributed Transform Coding(WT)	✓
[171]	✓	✓	✓	✓	Distributed Transform Coding(WT)	✓
[185]	✓	✓	-	✓	Distributed Transform Coding(WT)	✓
[189]	✓	✓	-	-	Temporal Compressive Sensing	✓
[110]	✓	-	✓	✓	Temporal Compressive Sensing	-
[191]	✓	✓	✓	✓	Temporal Compressive Sensing	✓
[190]	✓	✓	-	✓	Temporal Compressive Sensing	-
[192]	✓	✓	-	-	Spatial Compressive Sensing	-
[195]	✓	✓	✓	-	Spatial Compressive Sensing	-
[194]	✓	-	-	✓	Spatial Compressive Sensing	-
[193]	✓	✓	-	✓	Spatial Compressive Sensing	✓
[196]	✓	✓	-	✓	Spatial Compressive Sensing	-
[96]	✓	✓	✓	✓	Spatial Compressive Sensing	-
[199]	✓	✓	✓	✓	Spatial-Temporal Compressive Sensing	-
[198]	✓	✓	✓	-	Spatial-Temporal Compressive Sensing	-
[111]	✓	✓	✓	✓	Spatial-Temporal Compressive Sensing	✓
[197]	✓	✓	-	✓	Spatial-Temporal Compressive Sensing	-

sensor network characteristics. However, the complexity in the reconstruction procedure increases the latency parameter. Due to this latency, compressive sensing is not suitable for real-time applications.

5.4 Chapter summery

Data compression is one of the well-known data reduction techniques applied in sensor networks to cope with the resource restriction nature of these networks. In the last decades, we have witnessed a large number of researches proposed for this challenge. This chapter has reviewed and compared the recent advances in data compression approaches and their support for the various quality of service parameters. To this end, we divided existing approaches into lossy and lossless data compression categories. Then, we studied the performance of each method based on data quality, energy efficiency, scalability and latency parameters. Our studies revealed that lossy data compression techniques show better compression ratios which results in better energy efficiency. On the other hand, lossless solutions provide better data quality which comes at the cost of more energy consumption. In the category of lossless data compressions, dictionary and entropy based solutions represents a good compression efficiency, however, the existing spatial/temporal correlation in sensor networks are not utilized which limits their energy efficiency. Distributed source coding based methods achieves higher compression and energy efficiency, however, they are not scalable and loose their efficiency in dynamic networks. Our analysis on lossy data compression category showed that transform based compression and compressive sensing techniques propose higher energy efficiency and data compression ratios.

For the future work, the trade-off between energy and data quality, which is not properly addressed in most of the solutions proposed so far, can be studied. As an another issue, the application specific quality of service requirements are not properly considered in proposed compression solutions. Therefore, investigation on the application dependent compression techniques is also one of the open research questions.

On Analysis of Transform Basis in Compressive Sensing

6.1 Introduction

As described in Chapter 5, compressive sensing originates in the field of signal processing and has become a topic of frequent study in recent years. Compressive sensing ensures that sparse or compressible signals are accurately or approximately recovered from a number of linear projections of dimensions considerably lower than the number of samples required by the Nyquist-Shannon theorem [200, 201, 202].

Unlike classic compression techniques, compressive sensing simplifies encoding procedures while making the decoding procedure more complicated. This characteristic is well suited for wireless sensor network data gathering applications. It employs simple encoding on the resource-restricted sensor nodes and a complex decoding procedure at the powerful base station. Figure 6.1 illustrates the traditional data compression and compressive sensing techniques.

The compressible or sparse signal may be represented in different transform bases. Choosing the appropriate transform basis is essential in sparse representation, which can lead to less measurements and more accurate signal reconstruction during the recovery phase. To the best of our knowledge, the selection of an appropriate transform basis has not been adequately studied, as most existing compressive sensing techniques have focussed on optimizing reconstruction algorithms. This chapter classifies different signals based on their characteristics and investigates the assignment of an appropriate transform basis for each specific type of signal.

Figure 6.1: Traditional data compression vs compressive sensing

6.2 Preliminaries

Sampling is a process of conversion of continuous time signal into a discrete signal, and the reconstruction process involves interpolating the discrete time signals and producing a continuous time signal. Due to the nature of sampling, some information may be lost during the process. To avoid this information loss, Shannon and Nyquist proposed their famous theory, stating that for a continuous time signal which is perfectly bounded into a specific bandwidth frequency f , all signal information can be represented in a discrete domain as long as the minimum sampling frequency for the discrete domain is $2f$. Sampling with lower frequency results in aliasing after reconstruction, which leads to information being lost. However, the aliasing issue can be addressed through either increasing the sampling frequency to above $2f$ or employing an anti-aliasing filter to remove the high frequency parts [203, 204].

6.2.1 Sparsity

A sparse signal is a signal that exhibits sparsity, while a compressible signal can be well approximated with sparse representation by transforming to another

space where a small number of coefficients represent most power of the signals [205].

6.2.2 Compressive sensing

This section presents the mathematical description of compressive sensing theory as presented in [201, 205].

Let us assume that a discrete signal denoted by $X \in R^N$ represented by $N \times 1$ column vector has sparse representation in some basis (for example in fourier or wavelet basis). Considering the sparsity concept, this signal can be expressed in terms of its basis as follows:

$$X = \sum_{k=1}^N a_k \psi_k = \Psi a \quad (6.1)$$

, where Ψ is an $N \times N$ orthogonal basis matrix $\Psi = [\psi_1, \psi_2, \dots, \psi_N]$, ψ_i , $i = 1, 2, \dots, N$ and $a = [a_1, a_2, \dots, a_N]$ is an $N \times 1$ column vector of the coefficient sequence of X in Ψ domain.

Signal X is compressible or sparse in Ψ basis if most of the elements in a are zero. Compressive sensing theory states that if signal X is K -sparse on Ψ basis, it can capture and recover from M non-adaptive, linear measurements ($K < M \ll N$) with a certain restriction.

Compressive sensing theory also states that, rather than acquiring the entire signal and then doing compression, it should be possible to construct a sampling framework to capture only the useful information in the sparse signal to begin with. The sampled signal via compressive sensing is described as follows:

$$Y = \Phi X \quad (6.2)$$

, where $Y = [y_1, y_2, \dots, y_M]$ is the $M \times 1$ measurement matrix, $\Phi = [\phi_1, \phi_2, \dots, \phi_M]$ represents a $M \times N$ and each ϕ_i , $i = 1, 2, \dots, M$ is $N \times 1$ vector. It should be noted that Φ is a random matrix, which can be considered a second basis. Each elementary y_i in the measurement matrix is a product of X and vector ϕ_i from the sensing matrix. We can replace X with Ψa and rewrite Y as:

$$Y = \Phi X = \Phi \Psi a = \Theta a \quad (6.3)$$

,where $\Theta = \Phi \Psi$ is a $M \times N$ matrix.

Compressive sensing demonstrates that sparse signals can be recovered from M measurements if the process can satisfy the restricted isometric property (RIP) [206, 207]. The RIP states that the correlation between Φ and Ψ must be low (be incoherent) to restrict the number of measurements required to recover the signal. Coherence is a metric to measure the maximum correlation between any row of Φ and any column of Ψ and can be defined by μ the as follows:

$$\mu(\Phi, \Psi) = \max_{1 \leq k, j \leq N} |\langle \phi_k, \psi_j \rangle| \quad (6.4)$$

When comparing Φ and Ψ , if the relationship between them is weak, the condition is called incoherence. Formally speaking, matrix Θ of size $M \times N$ satisfies the RIP of order K if it is a minimum number such that

$$(1 - \delta_k) \|a\|_2^2 \leq \|\Theta a\|_2^2 \leq (1 + \delta_k) \|a\|_2^2 \quad (6.5)$$

, where $\delta_k \in (0, 1)$ is a restricted isometric constant (RIC). Equation 6.3 must hold true for all values of a where a $\|a\|_0 \leq K$ and $\|a\|_0$ is l_0 norm which shows the number of non-zero elements in a . A l_p norm of vector a is defined as follows:

$$\|a\|_p^p = \sum_{i=1}^N |a_i|^p \quad (6.6)$$

The RIP guarantees the exact recovery of X with a high probability if

$$M \geq CK \log \frac{N}{K} \quad (6.7)$$

, where N is the dimensionality of the original signal and C is the small positive constant. This formula is a conservative estimate of the lower bound on the number of measurements required to recover the original sparse signal. Since K represents the amount of important information inside the signal, we can conclude that the number of required samples to recover the signal is proportional to the information content of the signal.

The recovery of the signal X from Y is an NP hard problem, but it can be achieved through optimization. To do so we can use l_0 minimization as follows:

$$\hat{a} = \arg \min_{a \in R^N} \|a\|_{l_0} \quad s.t. \quad Y = \Phi X. \quad (6.8)$$

, where $\hat{X} = \Psi \hat{a}$, $\hat{X}$ is the recovered signal and $\hat{a}$ is the optimal estimation for a . Since l_0 minimization is computationally intractable, l_1 minimization is

widely used for compressive sensing signal reconstruction. We can recover the coefficients of sparse signal a by solving l_1 norm minimization as follows:

$$\hat{a} = \arg \min_{a \in R^N} \|a\|_{l_1} \quad s.t. \quad Y = \Phi X. \quad (6.9)$$

6.3 Sparse transform basis

Compressive sensing must utilize an appropriate transform basis to provide an accurate sparse representation for a compressible signal. The selection of these transform bases is very important and relates to the nature of the data or the signal to be compressed. In this section, we propose a classification for signals and then introduce an appropriate transform basis for each of them.

6.3.1 Classification

A signal is a physical quantity that is measurable. Various classifications of signals have been suggested based on the nature of the signal and the system that produces it. These include continuous versus discrete, periodic versus non-periodic, energy versus power, deterministic versus random, stationary versus non-stationary, and linear versus non-linear [208, 209]. In our study, we classify existing signals into a linear and non-linear signals. Linear here means that the natural frequencies of the signal are not dependent on the amplitude of oscillation. A signal can also be stationary or non-stationary. A stationary signal is one whose frequency or spectral contents change over time.

The proposed signal classification and transform basis allocation are illustrated in Figure 6.2.

6.3.2 Transform basis

Based on the proposed signal classification in Figure 6.2, we introduce wavelet, fourier and hilbert transformations and discuss which transform basis belongs to which type of signals.

Figure 6.2: Signal classification and appropriate transform allocation

6.3.2.1 Fourier transform

Fourier transform (FT) [209] identifies the frequency components in a linear time invariant signal. It can be described as follows:

$$X(f) = \int_{-\infty}^{\infty} X(t)e^{-2\pi i f t} dt \quad (6.10)$$

, where $f(t)$ is the time domain signal and $X(f)$ is the FT of $X(t)$ in the frequency domain.

If discrete measurements are taken, the discrete fourier transform (DFT) [208] of N uniformly sampled data points is used, which is defined as follows:

$$X = \int_{-\infty}^{\infty} X e^{-2\pi i f t} dt \quad (6.11)$$

The result of the DFT on the N-point input time series is an N-point frequency spectrum. Over discrete signals, the FT is decomposition on a discrete orthogonal fourier basis. In this case, the fast fourier transform (FFT) calculates the discrete fourier transform coefficients [210].

The fourier transform is suitable for a wide range of purposes, such as signal transmission and stationary signal processing. However, this transform is not appropriate for non-stationary signals because it cannot provide any information about signal frequency changes as a function of time for these signals. To cope with this limitation, the short time fourier transform (STFT) employs a fixed-size window to segment the signal into stationary parts but suffers from time or frequency resolution depending on its window size [208].

Most of the compressive sensing-based data-gathering approaches in wireless sensor networks use the FFT as their basis [211, 210]. This basis implements transformation over the entire length of the signal. In signal reconstruction, this transform basis provides no information about time resolution and as such fails to report damage locations since all possible time information is lost. The STFT makes location information available. Although it provides a time-frequency representation of a signal, there is a major drawback with respect to utilizing STFT due to the fixed size of its window width [212, 211].

6.3.2.2 Wavelet transform

The Wavelet Transform (WT) [213] is a time-frequency technique that overcomes the poor time/frequency resolution and fixed-size window limitations of the STFT. In contrast to the FT, the WT is well localized, and with it few coefficients are needed to represent local transient structures. Therefore, the wavelet basis provides sparse representation of piecewise regular signals, which may include transients and singularities [213, 209].

The variable window size employed by WT allows it to implement time-frequency transformation. When dealing with low- and high-frequency signals, it employs large windows and short windows respectively (corresponding to long and short time intervals). Therefore, the accurate location of transient signal can be determined, as well as fundamental frequency and its low-order harmonics [208].

The WT can be either continuous or discrete. The Discrete Wavelet Transform (DWT) is used for signal decomposition, while a Continuous Wavelet Transform (CWT) is used for spectral analysis [214]. The CWT is defined as follows:

$$CWT(a, b) = \int_{-\infty}^{\infty} X(t) \psi_{a,b}(t) dt = \int_{-\infty}^{\infty} X(t) \frac{1}{\sqrt{a}} \psi_{a,b}\left(\frac{t-b}{a}\right) dt \quad (6.12)$$

, where $X(t)$ is the original signal which is decomposed into a set of basis functions or wavelets.

Decomposing the CWT produces wavelet series. These series are highly redundant and their computation is costly and time-consuming. The DWT is a fast computation of WT which reduces time and resource cost [214]. The DWT decomposes signals into various resolution scales and is defined as follows:

$$C_{j,k} = 2^{-\frac{j}{2}} \sum_n x(n) \psi(2^{-j}n - k) \quad (6.13)$$

, where $\psi(n)$ is the wavelet function and $2^{-\frac{j}{2}} \psi(2^{-j}n - k)$ are scaled and shifted versions of $\psi(n)$ calculated based on scaling and shifting coefficients (j, k) .

Wavelet Multi Resolution (WMR) is a method that implements DWT with filters. It employs a high-pass filter to process rapidly changing details (high-frequency components) and a low-pass filter to find slowly changing features (low-frequency components). In contrast with the WMR, the Wavelet Packet Transform (WPT) is a method that decomposes both wavelet details and wavelet approximation components at each level. Wavelet packets inherit properties such as orthonormality and time-frequency localization from their corresponding wavelet functions [213, 214, 209].

There are number of basis functions that can be utilized as a mother wavelet for the wavelet transform, including Haar wavelet transform, Symlet wavelet transform, Coiflet wavelet transform, and Meyer wavelet transform [214, 209]. In terms of the signal characteristics and the attributes of specific wavelet transforms, each of them can be employed to represent sparsity of signal. The WT's ability to provide good frequency resolution for high-frequency signals and time resolution for low-frequency signals makes this technique a good basis for non-stationary signals that require time and frequency resolution.

6.3.2.3 Hilbert transform

Hilbert transform takes a function as an input and produces a function with the same domain. It is expressed as follows:

$$H[X(t)] = P.V. = \int_{-\infty}^{\infty} X(t)h(t-T)dT = \frac{1}{\pi} \int_{-\infty}^{\infty} \left(\frac{X(T)}{t-T}\right)dT \quad (6.14)$$

,where $X(t)$ is any real valued function, and $P.V.$ is the principal value of the singular integration. The Hilbert transform can be either continuous or discrete. The non-linearity and non-stationary nature of some signals make the Hilbert transform a powerful tool as the transform basis for signals with those properties. Hilbert transform decomposes a signal posteriori, enabling the extraction of the inner scales of each signal [209].

6.4 Analysis of transform basis: A case study

6.4.1 Structural health monitoring

Most of the environmental phenomena like the temperature data studied in first part of the thesis are usually non-stationary [82, 215, 11]. Therefore, we select this type of signal for evaluation and analysis of the effect of the transform basis on compressive sensing performance. To do so, we specifically consider structural health monitoring (SHM) applications [216, 217, 218]. In this case study, as illustrated in in Figure 6.3, wireless sensor nodes are deployed to survey a bridge, to detect the occurrence of possible damage, and to report the location and nature of the damage to the base station. These nodes measure bridge vibration. Due to the large amount of vibration data, compressive sensing is a good candidate to reduce the data. With compressive sensing, each sensor node encodes the data and then transfers the data to the base station. Upon receiving this data, the base station recovers and decodes the original data with a reconstruction algorithm.

During the encoding step, sensor nodes sample and compress data simultaneously. To do so, they employ a random measurement matrix to produce a linear combination of the sampled data. The sampling and encoding process carried out by each sensor is described as follows:

$$Y = \phi X \quad (6.15)$$

Figure 6.3: Sensor nodes deployment over the bridge

, where X denotes the sensing data captured in N consecutive time instances and Y is the $M \times 1$ measurement vector obtained by random projection from X to the random measurement matrix. After encoding, each sensor transmits the compressed data to the base station.

After receiving the encoded data, the base station must reconstruct signal X from Y . The reconstruction algorithm takes the measurement vector Y as the input and produces k -sparse vector X as an output. There are a variety of reconstruction algorithms that may be used for the recovery process. Since our objective is to evaluate the effect of the transform basis and not to improve recovery, we choose the CoSaMP [219] for this process.

6.5 Impact analysis of transform basis on compressive sensing

We analyse the effect of HT, WT (including Haar WT, Symlet WT, Coiflet WT, and Daubichi WT), and FT on the signal reconstruction accuracy. In order to compare the signal reconstruction accuracy, we use two metrics: SNR and normalized root mean square error (NRMSE). Figure 6.4 illustrates the comparison between HT, WT and FT in terms of SNR. As the figure shows, WT performs the best. Since WT is able to provide time and frequency information about the signal, it provides better sparsity than the other transform bases.

Figure 6.4: Comparing HT, WT, and FT in terms of SNR

Comparing them in terms of signal reconstruction error, as illustrated in Figure 6.5, shows that HT and WT have the worst and the best accuracy, respectively.

Figure 6.5: Comparing HT, WT, and FT in terms of signal reconstruction error

Figure 6.6: Comparing Haar WT, Symlet WT, Coiflet WT, and Daubichi WT in terms of SNR

Figure 6.7: Comparing Haar WT, Symlet WT, Coiflet WT, and Daubichi WT in terms of signal reconstruction error

Because WT shows the best performance, we further analyze a number of transforms sub-types. Figure 6.6 and 6.7 illustrate the comparison between Haar WT, Symlet WT, Coiflet WT, and Daubichi WT. As we see, most of the time Coiflet WT performs the best in terms of signal reconstruction error (measured as NRMSE) and SNR. Symlet WT performs the worst. However, when the measurement ratio is between 0.3 and 0.5, Haar WT shows better performance.

6.6 Chapter summery

Selecting an appropriate transform basis for compressive sensing plays a key role in sparse representation of the signals, which can lead to less measurements and more accurate signal reconstruction. However, each signal has specific characteristics that best matches a particular transform basis.

This chapter has presented a number of transform bases and analyzed their effect on compressive sensing performance in terms of SNR and signal reconstruction error. Since most signals acquired by wireless sensor networks are

non-stationary, we considered data collected from bridge monitoring application as a case study for this impact analysis. The comparison of simulation results using FT, HT and WT demonstrates that WT has the best performance when applying compressive sensing to this case study. Coiflet WT, Daubichi WT, Symlet WT, and Haar WT, the sub-families of WT, were also studied and analysed in terms of SNR and NRMSE.

Since most of environmental phenomena are non-stationary, we just studied the transform basis impact on those signals. However, there is other environmental data which are stationary and the impact of transform basis on those data (signals) can be investigated on the future work. Furthermore, in this chapter, we restricted the transform basis to the wavelet and fourier transforms while there are other transforms which can be studied in the future works.

Energy Efficient Distributed Compressive Sensing

7.1 Introduction

Recent compressive sensing solutions proposed in wireless sensor networks have proven useful in decreasing the number of measurements, however, they remain uncompetitive with the existing data compression techniques [220, 221, 222]. Since transmission cost is the dominant energy consumption parameter, compressive sensing methods should also consider minimizing the energy consumption by each measurement in terms of transmission cost.

In some proposed solutions, each sensor node produces its measurements, aggregates these with the measurements of other nodes and transfers them to the base station. Since sensor nodes may be far from each other, the transmission cost is high enough to negate the advantages of compressive sensing-based data gathering [223, 224, 225]. Utilizing distributed compressive sensing could represent an important step towards improving existing compressive sensing methods in terms of energy expense, bandwidth usage, data quality [226, 227].

Compressive sensing based data reconstruction is another research area in sensor networks, which addresses compressed data recovery procedure in the base station. Recent studies on compressive sensing employ linear programming and greedy algorithm to reconstruct the compressed signal [228]. Bayesian compressive sensing (BCS) [229] is another technique that uses statistical characterization of the signal to complement the conventional methods. It can provide better performance in terms of accurate data reconstruction, however, it is not widely used in the wireless sensor networks field [110]. TC-CSBP [110] is one of first researches which employs belief propagation (BP) technique to reconstruct the signal.

In this chapter, we propose a new distributed compressive sensing based data gathering approach which utilizes spatial-temporal correlation to improve the network performance in terms of energy consumption and data reconstruction accuracy. In the first step, we propose a spatial correlation based coalition formation algorithm to localize data collection of the sensor nodes. This approach leverages the sparsity distribution of signals in order to group spatially correlated sensor nodes into coalitions. Inside each coalition, we use the sparsity distribution of signals to define a utility function. This function later is used to minimize the number of active sensor nodes, which reduces energy consumption.

After coalition formation, a spatial temporal correlation based compressive sensing solution is proposed inside each coalitions. Proposed solution employs a block diagonal measurement matrix to produce a linear combination of sensor node readings. Block diagonal measurement matrix is structured in a way that balances the computation and communication load over the coalitions. This spatial-temporal correlation based compressive sensing is used inside each coalition to compress sensor node readings and transfer it to the base station. Upon receiving compressed data, the base station applies a novel two-step joint sparse signal recovery mechanism to reconstruct compressed data. In the first step, a joint sparse signal recovery is run to find the common sparsity profile among the coalitions. This recovery procedure then executes inside each coalition to achieve a common profile among sensor nodes. Utilizing this recovery algorithm helps to achieve better accuracy in data reconstruction while the number of required measurements is reduced.

We summarize our contributions in this chapter as follows:

- Proposing a new compressive sensing-based coalition-formation mechanism that includes distributed compressive sensing-based data gathering. This mechanism balances energy and quality of service.
- Introducing a sparsity distribution-based sensor node scheduling technique that minimizes the number of active sensor nodes.
- Proposing an adaptive measurement mechanism that adjusts the number of measurements inside each coalition according to the temporal sparsity level of the signal.
- Proposing a two-level belief propagation-based reconstruction algorithm that provides acceptable data quality in terms of data accuracy.

7.2 Preliminaries

7.2.1 Distributed compressive sensing

Standard compressive sensing methods can be implemented on single sensor node. However, sensor networks consist of many sensor nodes distributed over the network. These sensor nodes have spatial and temporal correlation among their readings. Given this distribution, distributed compressive sensing is proposed as a standard compressive sensing method for several sensor nodes [230, 231]. In this distribution, each sensor node employs compressive sensing theory to encode its reading and transfer it to the base station. For the reconstruction procedure, the base station utilizes joint sparse signal recovery to recover all encoded signals precisely.

Joint sparse signal (JSM) recovery [230] involves a set of signals that are jointly sparse. Each joint sparse signal can be represented by a combination of the common part and the innovative part [232]. There are three different joint sparsity models that can be used to recover the original signals. In this case we have an ensemble of signals with notation and measurement models as follows:

Each signal is denoted by $x_j, j \in \{1, 2, \dots, J\}$ where $x_j \in R^N$. For this model, it is assumed that there is a known sparse basis Ψ for R^N in which each x_j can be sparsely represented. On the other hand, each signal x_j is assigned a measurement matrix Φ which is $M_j \times N$. Based on this definition, a measurement signal for each signal j can be represented by $y_j = \phi_j \times x_j$.

In what follows, we introduce three different types of joint sparse models.

7.2.1.1 Joint sparse model Type-1

The joint sparse model type-1, called JSM1, states that all signals reported by sensor nodes have a common sparse component while each individual signal contains an innovative sparse component.

$$x_j = z_c + z_j \quad \text{with} \quad z_c = \Psi a_c, \|a\|_0 = K \quad \text{and} \quad z_j = \Psi a_j, \|a_j\|_0 = K_j. \quad (7.1)$$

where the z_c is common to the all of values for x_j and its sparsity level is the minimum sparsity level of all the signals in base function Ψ . Signal z_j is the specific innovative part of x_j and has its sparsity level in the same base function. In this model, joint signal recovery maximizes reconstruction of the common part to increase the reconstruction accuracy. When the proportion of the common part is far more than individual part, the reconstruction error decreases.

7.2.1.2 Joint sparse model Type-2

The Joint sparse model type-2, namely JSM2, assumes that signals have the same sparsity set profile, however, their coefficients are different. In this model we formulate each signal as:

$$X_j = \Psi a_j \quad (7.2)$$

, where a_j is non-zero in the common coefficient profile. This models data for the sensor nodes that are deployed in the environment and observe the same signal, but due to signal propagation effects, receive the signals with a phase shift.

7.2.1.3 Joint sparse model Type-3

There is some similarity between Joint sparse model type-3 (JSM3) and JSM1, but the common profile is no longer sparse. As with JSM1, we formulate the signal as follows:

$$x_j = z_c + z_j \quad \text{with} \quad z_c = \Psi a_c, \|a\|_0 = K \quad \text{and} \quad z_j = \Psi a_j, \|a_j\|_0 = K_j. \quad (7.3)$$

In this representation Z_c is not sparse. This model is applicable for environments in which several signals are produced from different sources with a background signal that is not sparse.

7.2.2 Belief propagation

Belief propagation (BP) [229] is an iterative message passing algorithm that can calculate the marginal distribution or find estimates such as the most probable assignment (MAP) in Bayesian networks [233, 234]. In the sparse signal recovery area, belief propagation is implemented on factor graphs and considered a fast decoder in Bayesian compressive sensing frameworks. This factor graph is a bipartite graph providing a visual representation of the sparse signal recovery procedure. It consists of two disjoint nodes: variable nodes and connection nodes connected through undirected edges whenever there is a dependency between these nodes. On the basis of belief propagation, these edges contain probability distribution functions on the variable nodes. There are only two directions for edges between variable and connection nodes [229]:

Figure 7.1: Message from variable node to the connection node in belief propagation model

1. Edges or messages from a variable node to the connection node:

These edges contain probability which is calculated by gathering all incoming edges (excepts edges coming from node c_m) and multiplying them, which is described as follows:

$$\mu_{x \rightarrow c}(x) = \prod_{x \in N(x) \setminus \{c\}} \mu_{v \rightarrow x}(x) \quad (7.4)$$

, where $\mu_{v \rightarrow x}$ shows the edges from node v to node x , $N(x)$ denotes the neighbor nodes of x and $N(x) \setminus \{C\}$ denotes the neighbor nodes of x except for node C . For an example, see Figure 7.1, which shows part of a

Figure 7.2: Message from connection node to variable node in belief propagation model

factor graph. The message from variable nodes X to the connection node c_m is given by:

$$\mu_{x \rightarrow c_m}(x) = \mu_{c_1 \rightarrow x}(x) \times \mu_{c_2 \rightarrow x}(x) \times \dots \times \mu_{(c_m-1) \rightarrow x}(x) \quad (7.5)$$

2. Edges from a connection node to the variable node:

This probability is computed by obtaining all the incoming edges to node C except the link from node x , multiplying them by C and finally finding the sum of all connected variable nodes except for node x . In general, the edges going from the connection nodes to the variable node can be described as follows:

$$\mu_{c \rightarrow x}(x) = \sum_{\sim \{x\}} c(N(c)) \prod_{v \in N(c) \setminus \{x\}} \mu_{v \rightarrow c}(v) \quad (7.6)$$

, where $N(C)$ is all variable nodes connected to C and σ is the sum of all connected variables except for x . $\mu_{v \rightarrow c}(v)$ shows the edges from variable node v to the connection node C . An example is seen in Figure 7.2, which shows part of a factor graph. The message from connection nodes C to variable node x_m is represented by

$$\mu_{c \rightarrow x_m}(x_m) = \sum_{(x_1, x_2, \dots, x_m - 1)} c_{(x_1, x_2, \dots, x_m - 1)} \times \mu_{x_1 \rightarrow c}(x_1) \times \mu_{x_2 \rightarrow c}(x_2) \times \dots \times \mu_{x_m \rightarrow c}(x_m) \quad (7.7)$$

7.2.3 Network model and assumptions

Our heterogeneous network consists of N static sensor nodes regularly placed inside a region as shown in Figure 7.3. It utilizes multi-hop communication to forward the readings to the base station. Each sensor node is assigned an integer ID denoted by n , which is in the range from 1 to N .

There are three types of sensor nodes in the network: normal nodes, coalition coordinator and base station. The normal sensor nodes have limited computational and storage capacity. Moreover, these sensor nodes are equipped with limited battery power, as such recharging or replacing their batteries is difficult and expensive. There are a few powerful but also resource-restricted nodes: the coalition coordinators, which have more memory and computation capabilities. The base station is a powerful node with significant computational and storage resources. The network follows a periodic monitoring scenario to gather data from the environment. In such a long-term monitoring application, the end user requires a continuous stream of environmental data although nothing specific is being detected. In this sort of applications, energy consumption and data accuracy requirements are more important than real-timeness.

The sensor nodes are grouped into coalitions. To form these coalitions, we consider the sparsity distribution of signals among sensor nodes and place all correlated sensor nodes in the same group as illustrated in Figure 7.4.

Figure 7.3: Network architecture

7.3 Overview of the proposed approach

In order to achieve an energy-efficient and quality-aware compressive sensing method, we introduce a distributed compressive sensing approach with spatial correlation among sensor nodes to group them into coalitions. The proposed coalition-formation method is represented by a block diagonal measurement matrix in which each diagonal entity corresponds to one of the coalitions. Upon forming coalitions, the proposed spatial-temporal correlation-based compressive sensing approach is implemented inside each coalition in

Figure 7.4: Coalition representation

order to schedule sensor nodes and encode their readings. Temporal correlation among sensor node readings allows our approach to adjust the number of measurements with regard to the temporally changing sparsity level.

After applying proposed compressive sensing solution inside each coalition and forwarding compressed data, the base station employs a two-step joint sparsity-based recovery algorithm to reconstruct the original signal. In the first step, it runs a joint sparsity model to find the common sparsity profile among the coalitions. In the second step, it calculates the common sparsity model within each coalition, which allows it to achieve data reconstruction with high accuracy and fewer measurements.

For a graphical view of the entire data-gathering procedure proposed in this chapter, see the flowchart in Figure 7.5. This flowchart will be described in more detail in the following sections.

7.4 Coalition formation

We utilize spatial correlation among sensor nodes and a base function distribution pattern in order to group sensor nodes into several coalitions. Using coalitions leads to an energy-efficient compressive sensing-based measurement system that puts correlated sensor nodes close to each other in the same coalition. Moreover, coalitions make our measurement system sparse, which means each measurement can be obtained through a linear combination of samples captured by a few sensors within a single coalition. Using this coalition structure, our approach decreases transmission cost. In addition, within each coalition we remove the redundancy among compressed measurements of different sensors, decreasing the number of data packets.

The compressive sensing-based coalition formation introduced here is inspired by the fact that the subject signals are distributed in the environment. On the other hand, the distribution of the signal elements is different for the sensor nodes based on their locations. Therefore, the sensor nodes observe the same signal with different resolution. On the other hand, the sparse representation of the signal can be shown in a sparsity base function. We apply the base function distribution over the network to define a cover parameter. This cover metric shows the degree of the base function covered by the sensor nodes. This parameter helps sensor nodes inside each coalition produce informative measurements, which results in accurate data recovery by the base station.

To make a concrete and accurate coalition, we define a utility function U based on cover, transmission and sensor node correlations. This function is used to evaluate the efficiency of the coalitions produced. To do so, U makes a trade-off between reconstruction accuracy (correlation between sensor nodes and base function distribution pattern) and data-transmission cost (distance between sensor nodes).

The following sections explain the prerequisites for the coalition-formation algorithm and conclude by presenting its procedure.

Figure 7.5: Approach overview flowchart

7.4.1 Measurement matrix

Since we divide the network into different coalitions, the data is gathered through different coalitions. Therefore, whole data may be divided into discrete blocks, in which each block is acquired via a local measurement operator. Assuming that we have divided the network into N_C coalitions and our signal X is partitioned into N_C blocks $X_1, X_2, \dots, X_{N_C} \in R^N$. Each block j which shows coalition j is assigned with a local measurement sub-matrix $\Phi^j : R^N \rightarrow R^{N_j}$. Each measurement sub-matrix Φ^j represents the measurement pattern in each coalition j .

Each of our measurement sub-matrixes Φ^j is assigned to one coalition matrix C , which permutes the entities of the original signal assigned for the specific coalition j . In order to assign each coalition j , through its measurement matrix Φ^j with the permuted signal coefficients x_j^C , we multiply C with X which produces $CX = X^C = [x_1^C, x_2^C, \dots, x_{N_C}^C]$ and finally we have:

$$Y = \Phi CX \quad (7.8)$$

, where:

$$\Phi = \begin{pmatrix} \Phi^1 & & & \\ & \Phi^2 & & 0 \\ & & \ddots & \\ 0 & & & \Phi^{N_C} \end{pmatrix} \quad (7.9)$$

The resulting matrix ΦC represents the distribution of sensor nodes inside each coalition and C is a coalition matrix.

Based on Equation 7.9 and sparse representation of the signal $X = \tilde{\Psi}a$ we obtain:

$$Y = \Phi CX = \Phi(C\tilde{\Psi})a = \Phi\Psi a \quad (7.10)$$

, where $\Psi = C\tilde{\Psi}$ is a permuted version of $\tilde{\Psi}$. As we have seen, our coalition matrix permutes the base functions. Through this permutation, each row in Ψ is a permuted row of $\tilde{\Psi}$ as follows:

$$\psi_i^T = \sum_{j=1}^N P(i, j) \tilde{\psi}_j^T \quad (7.11)$$

, where ψ_i^T and $\tilde{\psi}_j^T$ are the row vectors of Ψ and $\tilde{\Psi}$, respectively. if $P(i, j) = 1$, then the i^{th} row of Ψ is replaced with j^{th} row of $\tilde{\Psi}$.

After permutation with the coalition matrix, we have a proper pattern for each Φ^j to measure data from the sensor node inside each coalition j .

The coalition-formation mechanism proposed here leads to a block diagonal measurement matrix with the appropriate coalition matrix C relative to the location of the sensor nodes.

7.4.2 Utility function

The proposed coalition matrix C can be assigned to any coalition-formation method. During the coalition-formation procedure, we aim to make the best trade-off between energy savings and reconstruction accuracy. To do so, we introduce a utility function to evaluate the efficiency of the coalition-formation process. This utility function is defined based on the energy, correlation and cover degree parameters. The following sections explain these parameters in more detail and conclude by introducing the utility function.

7.4.2.1 Energy

We define the energy parameter for coalition-formation scenarios based on transmission, processing and measurement costs. Energy consumption is described as follows:

$$E_i = E_{comm} + E_{pro} + E_{meas} \quad (7.12)$$

, where $E_{comm}, E_{pro}, E_{meas}$ are communication, processing and measurement energy parameters, respectively. In this section, we focus on data transmission and measurement cost. We normalize these costs and replace energy requirements for the processing and measurement with the measurement costs. The transmission cost relates directly to the distance, while the measurement cost is influenced by the number of measurements. We replace the energy parameter with the normalized parameters of distance between two nodes inside the coalition and the number of measurements is represented as follows:

$$EC_o(i, j) = \frac{Dist(i, j)}{Dist_{max}} + \frac{M_i}{M_{max}} \quad (7.13)$$

, where $EC_o(i, j)$ is the energy cost in terms of the normalized distance between nodes i, j $Dist(i, j)$ and the normalized number of measurements taken by node i (M_i).

7.4.2.2 Correlation degree

Since sensor nodes are located close to each other and they sense the same signal with different resolutions, we are able to find different levels of spatial correlation among them. Existing compressive techniques usually ignore this correlation and transfer redundant compressed data, which costs energy. Our approach tries to remove the redundancy among compressed data, thus leveraging this spatial correlation. To do so, in the coalition-formation phase we force the algorithm to consider this correlation along with other parameters. When we have correlated sensors in the same coalitions, we remove redundancy in the compressed data using the algorithm described in the next section. The correlation metric among sensor nodes is defined as $Corr$.

$$Corr(i, j) = \frac{Cov(y_i, y_j)}{\sigma(y_i)\sigma(y_j)} \quad (7.14)$$

According to this formula, we define a binary variable CR which indicates whether two sensor nodes are sufficiently correlated. For this purpose, we rely on a user-defined correlation threshold $TH1$.

$$CR(i, j) = \begin{cases} 1 & \text{if } Corr(i, j) > TH1; \\ 0 & \text{if } Corr(i, j) \leq TH1. \end{cases}$$

7.4.2.3 Cover degree

Measured signals can be represented by a sparsity function distribution over the networks. These functions can also be grouped into one or more coalitions. The performance of our coalition mechanism in terms of recovered data accuracy greatly depends on the nature of the sparsity base functions. To evaluate this performance, we define a sparsity base cover degree (SCD) metric, which measures the degree of overlap between each coalition with base functions Ψ . Essentially, the SCD shows the energy overlap between the base functions and coalitions.

We define the SCD metric between each base function i and coalition Col_j as follows:

$$SCD(j, i) = \sum_{m \in Col_j} \psi^2(i, m) \quad (7.15)$$

, where m is a sensor node located in coalition j . This metric $SCD(j, i)$ indicates that measurements collected from coalition j contain information about

the measurements of other coalitions that cover the same base function i . Taking this coverage degree among different coalitions into account, we use a joint sparse signal recovery approach to recover the original signal. However, there are situations in which Ψ is covered only by one coalition. This means that for a K -sparse signal, the sparsity bases are contained in one coalition. Since this coalition is not known beforehand, we need to gather data from all coalitions, which is not energy efficient. Meanwhile, for accurate data recovery we must have $O(K \log N)$ measurements, meaning that we need $O(KN_C \log N)$ measurements from all coalitions. On the other hand, the redundant measurements from other coalitions that do not overlap with the base, do not contribute to improving data accuracy. If Ψ has even overlap among different coalitions, our data recovery accuracy is increased. To quantify the cover level of Ψ over coalitions, we define the maximum SCD as follows:

$$SCD_{max}(\Psi) = SCD_{max}(C\bar{\Psi}) = \max_{j,u} \sum_w \Psi^j(w, u), \quad SCD(\Psi) \in [0, 1] \quad (7.16)$$

SCD_{max} shows the maximum cover level of each coalition with the sparsity base, while Ψ^j is the base sub-matrix assigned to coalition j .

Forming coalitions such that the sparsity function can be recovered with several coalitions increases the quality of the recovered data. On the other hand, we must lower the number of correlated coalitions and the number of measurements to cut energy consumption. To find this balance, we must define a utility function as described in the next section. We discuss the number of coalitions and measurements and how SCD_{max} makes a trade-off between energy and quality in terms of the minimum number of measurements and data recovery accuracy.

7.4.2.4 Utility function formulation

In this section, we calculate a utility function U to evaluate the candidate coalition structures.

In the coalition-formation phase, we aim to build N_C coalitions to achieve minimum energy consumption while meeting data-quality requirements. Minimizing transmission and measurement costs, which is the main energy consumption parameter here, depends on SCD_{max} , M_i and $Dist_{ij}$.

The proposed utility function should evaluate adding a new sensor node to any coalition in terms of the trade-off between energy and quality. To do so, we define the utility function U for each combination of (n_i, Col_r) as follows:

$$U(n_i, Col_r) = CR(n_i) \times (ECo(n_i, Col_r) + \alpha SCD_{max}(n_i, Col_r)), \quad \alpha > 0 \quad (7.17)$$

, where $CR(n_i)$ defines the correlation level of sensor node i with other sensor nodes in Col_r . Regarding the $SCD_{max}(n_i, Col_r)$ parameter, adding a sensor node to different coalitions may change the value of SCD_{max} .

7.4.3 Coalition formation algorithm

In the model introduced in this thesis, the network consists of N sensor nodes $SN = \{n_1, n_2, \dots, n_N\}$ and $L = \{l_{n_i, n_j}\}$ is the set of all possible connections between sensor nodes. Two sensor nodes are considered to be connected if they are placed in communication range of each other. We assume that Ψ is known and all sparsity bases are normalized to 1 so that $SCD \in [0, 1]$. This algorithm aims to cut energy consumption, provided that the data-quality requirement is met. Adding a new sensor node to the coalition is a selection procedure that evaluates which coalition is the best candidate for adding a sensor node. To make a good selection we run an optimization algorithm on U .

Before describing the optimization algorithm, the SCD parameter must be redefined. During coalition formation, when we add a new sensor node to the coalition, SCD evaluates the effect of adding this node by assigning a weight for the link between existing node in the coalition and the new sensor node. Our coalition-formation method, in order to examine the effect of adding a sensor node to each coalition, defines the SCD factor as $SCD_{max}(l_{n_i, n_j}, Col_r)$ by considering the link, $l_{n_i, n_j} \in L$ and a given coalition Col_r . This link does not connect two nodes in the same coalition, which means $n_i \in Col_r$ and $n_j \notin Col_r$.

Now we redefine the utility function U for each combination of (l_{n_i, n_j}, Col_r) :

$$U(n_i, Col_r) = CR(n_i, n_j) \times (ECo(l_{n_i, n_j}) + \alpha SCD_{max}(l_{n_i, n_j}, Col_r)), \quad \alpha > 0 \quad (7.18)$$

$SCD_{max}(l_{n_i, n_j}, Col_r)$ here represents the maximum cover level when a new node is added to coalition Col_r using the link l_{n_i, n_j} .

Now we can formalize our optimization algorithm. The main goal in the improvement procedure is to lessen energy consumption and data recovery errors by adding new sensor nodes to the coalitions. When selecting an appropriate coalition for a new sensor node, the optimization algorithm examines the utility of adding the new sensor node to each coalition. For each coalition,

this utility is defined based on the link connecting the new sensor node to an existing node in the coalition. Running the optimization algorithm finds the coalitions that pare down utility function's link cost. We formulate this optimization as follows:

$$\begin{aligned} (n_i, l_{min}, Col_{r_{min}}) = \arg \min U(n_i, Col_r) = \arg \min [& CR(n_i, n_j) \times \\ & (ECo(l_{n_i, n_j}) + \alpha SCD_{max}(l_{n_i, n_j}, Col_r))] \quad (7.19) \\ \text{subject to } n_i \in SN, n_j \in Col_r, CR(n_i, n_j) \in \{0, 1\} \end{aligned}$$

Now let us describe the algorithm according to the optimization equations. To find a set of sensor nodes for each coalition, this algorithm uses Equation 7.19 to find a set of links such that the total U of the links is minimized. In the initialization step, we assume a set of candidate nodes and candidate links to be added to the coalitions defined by a set of SN and L . The algorithm then assigns each coalition coordinator node CC to one of the N_C coalitions. In addition, it defines SN_{Col_r} and L_{Col_r} as a set of sensor nodes and connections of coalition r , respectively. Along with the initialization step, the algorithm runs an iterative procedure where in each iteration, it allocates one sensor node to one of the coalitions. To do so, it first finds the utility function for all possible connections defined in L . It then runs the optimization function defined in Equation 7.19 and finds the minimum utility. However, it finds the utility for sensor nodes which satisfy the minimum correlation requirements. The output of this optimization is a connection link with minimum utility l_{min} . This link connecting (n_i, n_j) , adds sensor node n_i to the coalition of which node n_j is a member. Upon adding a new sensor node, the SCD parameter of the all links connected to the coalitions changes so that the link utility of the sensor nodes may change. It then removes this link and sensor node n_i from the list of candidate links and nodes. This procedure continues until all sensor nodes are assigned to the coalitions. Algorithm 7.1 represents this coalition formation procedure.

7.5 Data gathering inside coalitions

For the sensor nodes inside each coalition, we use a distributed compressive sensing technique to take measurements and recover the original signal. Before describing the data-gathering approach, in this section we take advantage of the SCD parameter to find the minimum number of active sensor nodes inside each coalition. The active sensor nodes inside coalition send their reading through a multi-hop network structure to the coalition coordinator.

Algorithm 7.1 Coalition formation

```

1:  $SN = n_1, n_2, \dots, n_N$ 
2: Define  $L = l_{ij}$  as a set of all possible links
3: Define  $N_C$  coalitions with one coalition coordinator
4: Define set of nodes  $N_{col_k}$  and links for each coalitions  $L_{col_k}$ 
5: for ( $P = 1; P \leq (N - N_C); P++$ ) do
6:   for ( $Q = 1; Q \leq |L|; Q++$ ) do
7:     Find  $L_{poss} = l_{ij}; i \in N_{col_k}, j \in SN$ 
8:   end for
9:   Calculate  $U(I_{ij}, Col_k)$ 
10:  Add  $n_j$  to  $Col_{k_{min}}$ 
11:  Add  $l_{min}$  to  $L_{Col_{min}}$ 
12:  Find  $(n_j, l_{min}, Col_{k_{min}})$ 
13:  Remove  $n_j$  from  $SN$ 
14:  Remove  $l_{min}$  from  $L$ 
15: end for

```

7.5.1 Number of active sensor nodes

In this section, we identify the number of sensor nodes required to be active for each coalition. This requirement helps to implement scheduling among sensor nodes and put some of the sensor nodes into sleep mode. Since the number of active sensor nodes measuring the signal is in direct relation with the number of measurements, we use Equations 6.4 and 6.7 (see Chapter 6) to find the number of active sensor nodes.

To do so, we first define a minimum threshold based on Equation 6.4 to measure the minimum data reconstruction quality. We then measure the probability of having coherence, as defined in Equation 6.4, higher than this threshold to make sure that the number of sensor nodes is enough to recover compressed signal with high accuracy. This probability is defined and represented as follows:

If all measurement matrixes ϕ_j are orthogonal and the sparsifying basis function $\tilde{\Psi}$ and coalition-formation matrix C are known a priori, then $\mu(\theta)$ is bounded as follows:

$$Pr[\mu(\theta) \leq O\sqrt{SCD_{max} \frac{N_C}{N} \log N}] = 1 - O(\frac{1}{N}) \quad (7.20)$$

Considering Equation 7.20 and Equation 6.7, we can define the maximum number of sensor nodes required for providing high quality data as follows:

$$N_{ANodes} = O(SCD_{max} K N_C \log^2 N) \quad (7.21)$$

Fine-tuning the coalition-formation matrix leads to a smaller SCD_{max} , which in turn produces a lower number of sensor nodes while still ensuring data quality. The number of active sensor nodes associated with each coalition is dependent on its SCD_{max} level. Coalitions, which cover more sparsity functions, are more informative. Therefore, we must gather more data from those coalitions.

The number of active sensor nodes for each coalition is represented as follows:

$$P_j = N_{Col_j} = \frac{SCD_{Col_j}}{SCD_{max}} N_{ANodes} \quad (7.22)$$

7.5.2 Compressive sensing based data gathering

We now construct a block diagonal measurement matrix using the spatial-temporal correlation among sensor nodes.

Let $SN_{Col_j} = \{1, 2, \dots, N_j\}$ denote the set of sensor nodes for the j^{th} coalition where P_j of these sensor nodes are randomly scheduled to be active. In contrast to existing compressive sensing methods, we define a new structure for the measurement matrix that is compatible with our coalition-formation method. We use a temporal block diagonal measurement matrix, Φ_t , to gather data. During each sampling instance, we gather spatial observations of all sensor nodes at time t and produce a discrete spatial signal X_t at this time. The temporal observation of all active sensor nodes together is a spatial-temporal signal $[X_1^{tr}, X_2^{tr}, \dots, X_{ST}^{tr}]$, where ST is a parameter representing the number of samples in each sampling round T . Each sampling period consists of T sampling instances equal to the Shannon-Nyquist rate. To reduce this number of sampling times, the base station adjusts the number of sampling times according to the signal sparsity level and defines a number of sampling points ST for each sampling period.

For each sampling period t , we consider Φ_t as a measurement matrix; Φ_t is a $P_j \times SN_{Col_j}$ matrix. The measurement vector Y_{Col_j} consists of ST_j sub-vector of ST sampling times such that $Y^{Col_j} = [Y_1^{tr}, Y_2^{tr}, \dots, Y_{ST_j}^{tr}]$ where each Y_i is a $P_j \times 1$ vector.

Inside each coalition, we utilize a block diagram measurement matrix that compactly represents several temporal measurement sub-matrices. Combining these spatial-temporal measurements together yields:

$$Y^j = Y_{Col_j} = \Phi^j X^j \quad (7.23)$$

$$Y^J = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{ST_j} \end{bmatrix} = \begin{pmatrix} \Phi_1 & & & \\ & \Phi_2 & & \\ & & \ddots & \\ & & & \Phi_{ST_j} \end{pmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{ST_j} \end{bmatrix} \quad (7.24)$$

, where for each $1 \leq t \leq ST$, Φ_t has P_j rows and SN_{Col_j} columns. At the end of each sampling round, every sensor node transfers its measurement vector Y_j to its neighbor node. The neighbor node receives the data and forwards it to the coalition coordinator.

Introducing this temporal block diagonal measurement inside each coalition provides energy-balanced data aggregation within the coalitions. For instance, consider the sample coalition in Figure 7.6, where active sensor nodes forward their compressed data through active neighbor nodes to the coalition coordinator. Existing compressive sensing-based data collection approaches introduce unbalanced data transmission, which leads to communication and data overhead in the network. In our approach, each sensor node transfers a total of M number of measurements where $M_j = m_1 + m_2 + \dots + M_{SR_j}$. For each sampling time t , sensor node i sends its measurement y_i, t to the next node j .

For each time t , we represent the data-gathering process as follows:

$$y^t = \Phi^t x^t \quad (7.25)$$

where $\Phi^t = [\phi_1, \phi_2, \dots, \phi_{N_{Col_k}}]$ is the measurement matrix of size $P_{Col_k} \times N_{Col_k}$ and y^t is the compressed measurements of signal x^t .

Each element y_i of measurement vector y^t is a linear combination of all sensor node readings inside the coalition, which is weighted with a row of the measurement matrix. This representation of each measurement gathered from active sensor nodes at time t can be represented as follows:

$$y_i^t = \sum_{j=1}^{N_{Col_k}} \phi_{ij}^t x_j^t, \quad i \in 1, 2, \dots, P_{Col_k} \quad (7.26)$$

Using this representation we balance energy usage through an equal number of transmissions among sensor nodes inside each coalition. According to

Figure 7.6: An Example Gathering Scenario

the proposed temporal representation, each sensor node is able to add its measurements $\{\phi_{ij}^t\}_{i=1}^{P_{Col_j}} x_j^t = \phi_j^t x_j^t$ to the measurements of other sensor nodes separately to produce each measurement. To do so, each P_{Col_j} active sensor node inside the coalition locally multiplies its readings by measurement vector ϕ_j^t and produces its measurement vector M^t . It then waits until it receives measurement vectors from other sensor nodes that forward their data to the coalition coordinator (CC) through this sensor node. Upon receiving them, this sensor node aggregates its measurement vector with the received measurement vector and forwards it to the CC. This procedure is depicted in Figure 7.6.

One question not yet answered concerns the means of routing sensor node measurements to the coalition coordinator. In the next section we describe how sensor nodes form a routing tree and forward their measurements.

7.5.3 Data gathering trees

Inside each coalition, our network consists of P_j active nodes connected through a set of links L . For each time stamp, every sensor node reads environmental data and transmits it to the coalition centre. Since not all sensor nodes

have access to the coalition centre, they forward their readings with multi-hop communication. In the compressive sensing technique, each coalition centre requires gathering only m projections from these sensors instead of all readings from sensor nodes where $m \ll n$.

As indicated above, each measurement is the weighted sum of active sensor node readings with non-zero coefficients. Our temporal measurement matrix consists of m_j rows and P_j columns where each column is assigned for one active sensor node and each row shows one of the projections. Since each row shows the selected sensor nodes whose readings contributed to the weighted sum m_j , we can consider each row as a data-gathering tree. In this regard, we define one of the sensor nodes with direct access to the coalition centre as a root of this tree. Therefore, having m_j projection we obtain m_j data-gathering tree. To distribute the non-zero coefficients more evenly in matrix ϕ and make each projection as sparse as possible, the number of non-zeros in each row of matrix ϕ is set at $\left\lceil \frac{m_j}{P_j} \right\rceil$ such that none of the columns in ϕ has all-zero entries.

We now try to find m_j data-gathering trees, each of which corresponds to one weighted sum of selected sensor nodes such that it minimizes the energy consumption by minimizing communication links. The following section describes how each projection node builds its measurement tree.

7.5.3.1 Building data gathering tree

Selecting m_j projection nodes and assigning a data-gathering tree based on the optimized random basis matrix are the main objectives of this section. To select m_j projection nodes, the coalition coordinator randomly selects m_j nodes that are one hop from it, and fixes them as roots of m projection trees. Later, during the data-gathering procedure, the coalition coordinator can change the role of projection nodes and notify these nodes by sending update messages.

Each projection node initializes one of m_j data projection procedures using the proposed algorithm. As mentioned above, to build the data-gathering tree for each projection we construct an optimized basis matrix and assign each row of this matrix to one of the projection trees. Each column of this projection matrix is assigned for one of the active sensor nodes, while each row is assigned to one projection node. Non-zero coefficients in each row represent the nodes that participate in that projection.

To make each random projection as sparse as possible, this algorithm sets the number of non-zero entities in each row of the measurement matrix ϕ as

equal to $\frac{m_j}{P_j}$, on the condition that none of the columns in ϕ has all-zero entries. The proposed measurement matrix has been generated in advance and stored in each sensor node. Now we demonstrate building data projection trees within each coalition.

In each coalition, the coalition coordinator gathers m_j weighted measurements from all active sensor nodes inside the coalition. Each of these m_j measurements is collected by one of the measurement nodes. To do so, each row of our measurement matrix represents the data-gathering tree assigned to one measurement node and each binary entity in this row vector shows whether this node is participating in the current measurement. Each measurement node uses this measurement matrix to detect sensor nodes that will participate in the measurement. It then builds a data-gathering tree based on the minimum spanning tree algorithm to develop a route between participating nodes and the measurement node.

The measurement node initiates the data-gathering tree by introducing itself as a root. It then finds all participating nodes located within one hop and adds them to the tree. For other nodes, it finds the closest node and follows the shortest path method to connect that node to the tree. This procedure repeats until all participating nodes join the tree. When the tree is complete, each sensor node takes its sample x_i , multiplies it by its coefficients, adds it to the data received from its child nodes and finally transmits the weighted sum to its parent node. At the end of this procedure, the measurement node receives all the weighted measurements from the tree and forwards them to the base station. Since the measurement matrix consists of m_j rows, we will have m_j trees, each of which gathers one of the measurements through the assigned measurement node.

Upon reading a sample from the environment, each sensor node in each tree multiplies its reading by its coefficient, adds it to the data received from its child nodes and sends the weighted sum to the parent node. Repetition of this procedure in the tree terminates when the measurement node receives weighted sums from the child nodes and then sends these readings to the coalition coordinator. In this way, all sensor nodes transmit only one packet to the coalition coordinator. Consequently, for m data-gathering trees, the network transmits m packets.

7.6 A two steps signal recovery procedure

The base station is responsible for recovering the original signals from the measurements received from the coalitions. Considering several coalitions, signals

gathered by the coalition coordinators are correlated in spatial and temporal domains. In essence, we have two types of correlations: spatial-temporal correlation inside each coalition and spatial correlation among coalitions. The distributed sparse signal approach proposed here engages these correlations and suggests a two-step joint sparse signal recovery process. In the joint sparse signal recovery concept [235, 236], this correlation can be defined based on location or amplitude of non-zero coefficients of the signal. In our approach, correlation among different coalitions or sensor nodes is depicted as a similarity level and is defined based on the location of the non-zero coefficients as follows:

$$SL = \frac{Com}{K} \quad (7.27)$$

, where Com and K , respectively, are the total number of common non-zero coefficient locations among different coalitions or sensor nodes and the total number of non-zero coefficients of the signals (sparsity level).

At the end of each data reconstruction procedure, the base station calculates and finds the common sparsity profile CSP_{Col} among different coalitions. Coalitions that satisfy minimum similarity requirements are considered in the joint sparse signal recovery process; otherwise, their recovery proceeds separately.

Based on this SL definition, in this section we develop a two-step joint sparse signal recovery algorithm which uses the spatial-temporal prior information to reconstruct the original signal. Using this algorithm, our approach decreases the number of measurements while increasing the reconstruction accuracy.

Our signal recovery procedure is performed both inside the coalitions and among the coalitions. In the first step, the base station runs the joint sparse signal recovery algorithm among coalitions and finds their common sparsity profile (CSP_{Col}). In the second step, using CSP_{Col} as an input for the reconstruction algorithm, it runs joint sparse recovery inside each coalition to complete the common sparsity profile of each coalition (CSP_{Col_j}). Since the sensor nodes inside each coalition are more highly correlated, running the second round yields other parameters of the common sparsity profile. Upon finding the common profile among sensor nodes (CSP_{Col_j}), the base station runs the individual recovery procedure to find individual parts. For the whole recovery process, the base station exploits the belief-propagation-based recovery algorithm introduced in the following section. Figure 7.2 summarizes the proposed recovery procedure.

Algorithm 7.2 Two steps joint signal recovery

-
- 1: Find SL among different coalitions
 - 2: Find SL_{min}
 - 3: Run spatial-temporal joint recovery inside coalitions with SL_{min}
 - 4: **for** ($J = 1; J \leq N_{Col}; J++$) **do**
 - 5: **if** $SL_{Col_j} > \theta$ **then**
 - 6: Use the common sparsity profile from coalition with SL_{min} as input
 - 7: **end if**
 - 8: Run spatial-temporal joint recovery algorithm
 - 9: **end for**
-

7.6.1 Sensor node signal model

The spatial-temporal signal model, which we aim to resolve, is represented as follows:

$$Y = \Phi X + E \quad (7.28)$$

Since we use a distributed compressive sensing technique, we can change multiple measurement representation to the single measurement representation. As discussed in Section 7.2.1.2, the common sparsity part is the same among all sensor nodes. Therefore, for this part, our multiple measurement vectors are reduced to a single vector and we have a common measurement basis matrix and a single-state vector for all the sensor node readings.

The T vectors of X , which represents the signal coefficients in T time stamps, have a common sparsity profile that is constant in time. We represent this support set as follows:

$$s_j^t = s_j, \quad \forall t \in [T] \quad (7.29)$$

Each signal X^t consists of a small number of large elements and a large number of small elements.

To distinguish between large and small elements, we define a status vector of common support, set as a binary vector, to indicate whether each coefficient of this signal is large or small.

$$s_j = \begin{cases} 0 & \text{if } x_j^t \text{ is small;} \\ 1 & \text{if } x_j^t \text{ is big.} \end{cases}$$

According to this representation, each state variable for support set is a binary value which takes 0 value when the corresponding element has a small magnitude, and value 1 when the element has a large magnitude.

We utilize a Gaussian distribution to associate the probability function for each coefficient of the signal. For $s_j = 1$, we select a high-variance zero-mean Gaussian distribution, and for $s_j = 0$, we choose a low-variance zero-mean Gaussian distribution. Therefore, the conditional probability of x_j^t on s_j can be represented as follows:

$$p(x_j^t | s_j) = \begin{cases} p(x_j^t | s_j = 0) \sim N(\mu_0, \sigma_0^2); \\ p(x_j^t | s_j = 1) \sim N(\mu_1, \sigma_1^2) \end{cases}$$

The coefficients of the signal are correlated as well and can be represented as follows:

$$\theta_j^t = (1 - \alpha)(\theta_j^{(t-1)} - \zeta) + \alpha\omega_j^t + \zeta \quad (7.30)$$

, where ω_j^t, α and ζ represent the driving process, correlation degree and means of the process, respectively.

7.6.2 Problem formulation

We consider sparse signal recovery to be a Bayesian inference problem. Therefore, the signal can be recovered by maximizing the posterior probability (MAP) estimation [236] as follows:

$$\hat{X}, \hat{S} = \arg \max_{X, S} f(X, S | Y) \quad (7.31)$$

This representation is the signal-wise MAP estimator which we use as the component-wise MAP estimator to solve the problem through the sum-product algorithm as follows:

$$\hat{X} = \arg \max_{X, S} \Pi f(X | Y, S) f(S | Y) \quad (7.32)$$

Replacing f with the probability function, we obtain:

$$\hat{X} = \arg \max_{X, S} \Pi P(X | Y, S) P(S | Y) \quad (7.33)$$

Figure 7.7: Belief propagation based recovery procedure

During the recovery process, the base station pursues two main goals:

1. Estimating the signal elements.
2. Detecting the support set of the signal.

In order to detect the support set, the base station utilizes the posterior distribution of s^i to find the following hypothesis test:

$$\frac{P(s^i = 1|X)}{P(s^i = 0|X)} \quad (7.34)$$

To solve this problem, we use a factor graph representation [237, 234]. As presented in Figure 7.7, this is a bipartite graph $G = (VN, CN, E)$ consisting of variable nodes VNs , connection nodes CNs and edges Es . As described in the Section 7.2.2, this factor graph representation allows us to easily find the marginal probability distribution. To build this graph, we consider variable nodes and connection nodes and connect these nodes through the undirected

edges when a connection node depends on a variable node. This representation allows us to calculate the marginal distribution functions by message-passing between the variable nodes and connection nodes.

Our graph consists of two sub-graphs with common nodes at $VN2$ containing signal coefficients $\{z_j^t\}$: The first sub-graph (sub-graph 1) computes the approximate posterior marginal distribution $\{z_j^t\}$. Selecting the appropriate probability distribution functions in this section guarantees a low-sparsity solution. The second sub-graph (sub-graph 2) provides an approximation of the signal elements and estimates the signal sparsity level.

Below we describe the reconstruction procedure of this graph in an abstract way.

7.6.3 A belief propagation-based recovery algorithm

Greedy and linear programming are the most common recovery algorithms utilized in wireless sensor networks [226, 227]. Considering the statistical characteristics of the signals, Bayesian recovery approaches can complement conventional compressive sensing methods based on linear programming or greedy algorithms. The recovery method proposed here is a joint sparsity-based compressive sensing technique that considers this property.

In our recovery approach, the base station employs Bayesian inference to build a probabilistic model of the signals and then applies BP as a decoding method to recover the sparse signal. We organize the network activities into several rounds. This means that the base station runs recovery algorithms at fixed time intervals. The sections below focus on the recovery algorithms run by the base station to reconstruct the sensor node measurements. As stated above, we run this procedure two times, first to find the common profile among coalitions and second to use this common profile as an input in the joint recovery algorithm to find the common sparsity profile inside each coalition. Finally, the individual parts are reconstructed using the common recovery part results.

7.6.3.1 Signal Bayesian prior model

The proposed BP-based approach reconstructs the signal by graphical representation. Before describing this representation, we must find the prior probability distribution of signal elements considered as an input in the factor graph. Each state variable is presumed to be a Bernoulli random variable with $P(s_j = 1) = \beta$. Since the signal is K -sparse, we can assume $\beta = \frac{K}{N}$. Therefore, the prior probability distribution of each state variable is as follows:

$$P(s_j) = \begin{cases} \beta = \frac{K}{N} & \text{if } s_j = 1; \\ 1 - \beta = 1 - \frac{K}{N} & \text{if } s_j = 0. \end{cases} \quad (7.35)$$

Now we use the factor graph representation to implement our approach. For sub-graph 1, two variable nodes and one connection node are defined: state variables $VN1$ and signal coefficients $VN2$ are the variable nodes, while $CN1$ is the connection node linking state variables S and signal elements Z .

By running the BP technique over this graph, the edges from the variable node to the connection node provide the belief about the current estimation of the signal coefficients. This belief is later used to update the probability of the signal sparsity level. Therefore, $VN1$ sends the distribution of each state variable $p(s_j)$ to $CN1$.

This part of the sub-graph is shown in the left-hand side of Figure 7.7. Since throughout the graph, most of the links between the nodes are bidirectional, for each part of the sub-graph we show the direction based on the direction of messages for the current situation. As indicated in Section 7.2.2, the message from the variable nodes to the factor nodes follows Equation 7.4.

The messages going from variable node $VN1(\{s_j\})$ to connection node $CN1(\{p(x_j^t|a_j^t)\})$ are a Gaussian-Bernoulli distribution function that will be parametrized by $\vec{\pi}$. According to Equation 7.4 this message takes the following form:

$$\mu_{VN1 \rightarrow CN1}(VN1) \propto (1 - \vec{\pi}_j^t)(1 - s_j) + \vec{\pi}_j^t s_j \quad (7.36)$$

$CN1$ finds the Gaussian distribution of z_j^t by marginalizing signal elements based on state variables as follows:

$$CN1(z_j^t|S_j) \sim p(z_j^t|s_j) \quad (7.37)$$

The edges from the connection nodes $CN1$ to the variable node $VN2$ can be calculated as follows:

$$\mu_{CN1 \rightarrow VN2}(VN2) = \sum_{\sim VN2} CN1(N(CN1)) \prod_{v \in N(CN1) \setminus \{VN2\}} \mu_{v \rightarrow CN1}(v) \quad (7.38)$$

, where $N(CN1)$ represents all variable nodes connected to $CN1$ and $\sum$ is the sum of all connected variables except for $VN2$. $\mu_{v \rightarrow CN1}(v)$ shows the edges going from variable node $VN2$ to connection node $CN1$. BP indicates

that the edges from the connection nodes to the signal elements nodes represent the belief about sparsity level. This edge essentially carries two Gaussian distributions for the zero and non-zero coefficients of the signal. Therefore, $CN1$ calculates the mixture Gaussian distribution of the signal elements and sends the parameters of this distribution to the $VN2$. Upon receiving these parameters, $VN2$ calculates the prior probability distribution of z_j^t as follows:

$$VN2 = p(z_j^t | s_j) = \lambda \times \mathcal{N}(\zeta_1, \phi_1^2) + (1 - \lambda) \times \delta(z_j^t) \quad (7.39)$$

, where $\delta(z_j^t)$ is a Dirac distribution function [238] and $\int \lambda(z) dz = 1$.

The second sub-graph, shown on the right-hand side of this factor graph, has two variable nodes and one connection node: $VN2$ and $VN3$ are the variable nodes, while $CN2$ is the connection node linking $VN2$ and $VN3$. This sub-graph is responsible for calculating the marginal distribution of the signal elements. As Figure 7.6 demonstrates, this distribution can be calculated by multiplying all incoming messages to the variable nodes.

Connection node $CN2$ receives all these incoming edges from the variable node $VN2$. For this connection node, the edge coming from $VN2$ provides belief about the current state of the signal elements. The incoming edges are the mixture of Gaussian densities. Each member of $VN2$ broadcasts its Gaussian density to all connection nodes involved in the measurement and $CN2$ is a delta function node that provides the relationship between signal observation variable node $VN3$ and signal coefficient variable nodes $VN2$. This connection node is defined as follows:

$$CN2(Z^t, y_j^t) = \delta(y_j^t - \sum_j a_j^t z_j^t), \quad z_j^t \in Z^t \quad (7.40)$$

For each variable node z_j^t , variable node y_j^t multiplies the delta function of connection node with other density functions received from variable nodes y_j^t to find the marginal distribution of each variable as follows:

$$p(z_j^t | Y) = CN2(Z^t, y_j^t) \times \prod_{k \neq j} \mu_{z_k^t \rightarrow y_j^t}(z_k^t) \quad (7.41)$$

This marginal distribution is calculated in each iteration of the algorithm until it converges. Since the signal coefficients are independent, the joint distribution of the signal Z can be calculated as follows:

$$p(Z^t | Y) = \prod_{j=1}^N p(z_j^t | Y) \quad (7.42)$$

Therefore, each connection node gathers all incoming edges (Gaussian densities), which are simply the multiplication of all incoming links to the connection node.

7.6.3.2 Support set detection

Another goal of the proposed graph is to find the support set of the signal. In this regard, the base station must find the posterior distribution of the state variables from sensor observations by calculating the following probability ratio:

$$SSet = \left\{ j : \frac{p(s_j=1|Y)}{p(s_j=0|Y)} \geq 1 \right\} \quad (7.43)$$

Factorizing over z_j^t this probability ratio becomes:

$$\frac{p(s_j = 1|Y)}{p(s_j = 0|Y)} = \frac{\int p(s_j = 1|Y, z_j^t) p(z_j^t|Y)}{\int p(s_j = 0|Y, z_j^t) p(z_j^t|Y)} \quad (7.44)$$

As shown in Figure 7.7, part of the graph related to this equation consists of two connection nodes $CN2$ and $CN3$ and three variable nodes $VN1$, $VN2$, and $VN4$. In each iteration of the algorithm, variable nodes $VN3 = Y$ pass their marginal distribution values to the variable nodes $VN2 = Z$. The variable node $VN2$ calculates the Gaussian density of all incoming edges by multiplying these messages. The link between variable node z_j^t and connection node $CN1$ then passes the parameters of this function to the connection node. These edges provide the second parameter of the aforementioned probability ratio. The first part of this ratio has been calculated through connection node $CN3$.

Now we must find the beliefs propagated from the variable node z_i^t to the factor node $CN3 = p(z_j^t|a_j^t, s_j)$. We define this edge from $VN2$ to $CN3$ as follows:

$$\mu_{VN2 \rightarrow CN1}(VN2) = \mu_{VN2 \rightarrow CN3}(VN2) = \mathcal{N}(z_j^t | \hat{r}_j^t, \tau^r(t)) \quad (7.45)$$

$VN2$ builds the model of signal based on the signal parameters a_j, s_j . Therefore this variable node itself consists of two sub-variables. First, we calculate the edge from connection node $CN1(p(z_j^t|a_j^t, s_j))$ to the variable node $VN1$, which is a type of posterior activation probability.

$$\mu_{CN1 \rightarrow VN1}(VN1) \propto (1 - \frac{1}{\pi_j^t}) + \frac{1}{\pi_j^t} s_j, \quad (7.46)$$

The edges from $CN3(p(x_j^t|a_j^t, S_j))$ to variable node $VN4(a_j^t)$ use the sum product rules, which yields:

$$\mu_{CN3 \rightarrow VN4}(VN4) = \mathcal{N}(a_j^t | \overleftarrow{\xi}_j^t, \overleftarrow{\psi}_j^t) \quad (7.47)$$

Now we propagate current beliefs between consecutive time frames, represented by the connection from a_j^t to a_j^{t-1} or a_j^{t+1} . In this phase, we find the common sparsity profile and the temporal correlation of the coefficients. To do so, $VN1$ calculates the posterior distribution of the support set as follows:

$$P(s_j|Y) \propto \mu_{p(s_j) \rightarrow s_j}(s_j) \prod_{t=1}^T \mu_{p(z_j^t|a_j^t, s_j) \rightarrow s_j}(s_j) \quad (7.48)$$

In addition, the posterior distribution of the coefficient is represented as follows:

$$P(a_j^t) \propto \mathcal{N}(a_j^t | \overleftarrow{\eta}_j^t, \overleftarrow{\kappa}_j^t) \mathcal{N}(a_j^t | \overrightarrow{\eta}_j^t, \overrightarrow{\kappa}_j^t) \mathcal{N}(a_j^t | \overleftarrow{\xi}_j^t, \overleftarrow{\psi}_j^t) \quad (7.49)$$

A MAP estimation of the support set can then be derived from the above distribution.

7.7 Performance evaluation

In order to evaluate our solution, we compare its distributed compression nature with centralized compressive sensing methods. We then compare our solution with other compressive sensing methods.

7.7.1 Assumption

For simulation, we have three different kinds of nodes: base station, coalition coordinator and sensor nodes. The base station groups these sensor nodes into different coalitions and assigns one coalition coordinator for each coalition. The coalition coordinator is responsible for transferring the coalition information to the base station. The base station is responsible for forming coalitions and running the optimal scheduling algorithm, which schedules communication in the network. Thus, there are no collisions in the network within or between coalitions. We transfer data by multi-hop communication, and each sensor node knows its geographical location and its connectivity radius. Two nodes are connected if and only if they are located in the communication range of each other. Coalition coordinator nodes are powerful nodes that act as aggregation centres.

We use a real dataset collected by Lausanne Urban Canopy Experiment (LUCE) wireless sensor network deployment at EPFL LUC[113]. This dataset gathers real temperature data using 64 ambient temperature sensor nodes. The sensor type is SHT75 sensor Sensirion. This dataset is sparse which made it as a very interesting dataset to evaluate recent compressive sensing solutions[82][239][240][241].

Our simulation is implemented in a MATLAB environment and we have used the toolbox provided for Bayesian compressive sensing in [236].

7.7.2 Comparing distributed and individual compressive sensing

The first part of the performance evaluation section investigates and compares distributed compressive sensing and individual compressive sensing approaches.

Typical signals measured by wireless sensor networks are sparse signals consisting of a common sparse component and different innovation components. Reconstructing these sorts of signals using joint sparse signal recovery models is more efficient than individual signal recovery models. It ultimately reduces the energy consumption and prolongs network lifetime.

For this simulation, we change the number of common (k_c) and innovative parts (k). As depicted in Figure 7.8, distributed compressive sensing provides accurate reconstruction with fewer measurements. On the other hand, increasing the number of common sparsity parameters provides better reconstruction accuracy. When the number of the common sparsity level is minimized, its performance approaches that of a separate compressive sensing method.

Figure 7.8: Reconstruction accuracy in terms of different common sparsity levels

7.7.3 Comparing with other compressive sensing based methods

In this section we compare the proposed spatial-temporal compressive sensing approach with the existing data compression techniques. The approaches used in the comparison are Bayesian compressive sensing (BCS) [242], clustered spatio-temporal Bayesian compressive sensing (STBCS) [243], temporal belief-propagation based compressive sensing (TBCS) [110], OMP based compressive sensing [244] and spatial Bayesian compressive sensing (SBCS) [245].

7.7.3.1 Data accuracy

In order to measure the data accuracy, we first compare the accurate reconstruction percentage for the different data compression values. As Figure 7.9 indicates, our approach provides accurate data reconstruction with a minimum number of measurements. The main reason for this performance is the value of information gained using the coalition-formation method. Our approach tries to group sensor nodes based on their sparsity similarity, which leads to transmitting more information with a lower number of measurements. However, increasing the compression ratio decreases the gap between our approach and other methods. On the other hand, our approach adjusts the number of measurements based on the signal sparsity level, which removes redundant data transmission. As it can be seen in Figure 7.9, for more compression ratios the reconstruction accuracy is almost the same.

Figure 7.9: Reconstruction accuracy percentage for different compression ratios

Figure 7.10: Reconstruction accuracy percentage for different SNRs

An important parameter in the reconstruction procedure is the level of SNR. Figure 7.10 represents the reconstruction accuracy based on the SNR parameter. For lower SNR values, our approach represents better reconstruction accuracy. When increasing the SNR value, the STBCS model provides a similar level of accuracy but still lower than in our method. At SNR levels higher than 19, our method produces almost perfect reconstruction. Among other methods, the STBCS technique is the only one which shows comparable performance.

Figure 7.11: Number of data transmission versus number of sensor nodes

7.7.3.2 Number of transmissions

One of the most important parameters affecting energy consumption is the number of data transmissions. The number of sensor nodes participating in data transmission is used to measure the number of transmissions. Figure 7.11 represents the number of transmissions for different compression techniques by changing number of sensor nodes.

For a network with a lower number of sensors, our approach has a minimum number of transmissions, but this number is comparable with STBCS and SBCS approaches. As the number of sensor nodes increases, the difference in the number of transmissions becomes more significant. This graph also demonstrates the scalability of our method. Our approach localizes sensor nodes based on the similarity of their sparsity levels in separate coalitions. For each coalition, it then curtails the number of transmissions by lessening the number of active sensor nodes participating in data transmission and adjusting the number of measurements to the sparsity level. Therefore, our approach has a minimum number of transmissions.

7.7.3.3 Energy consumption

In this section we compare energy consumption parameters among the different methods. As we can see in Figure 7.12, our approach provides the best energy resource consumption performance among the approaches compared. Our approach adapts both the number of measurements and the number of active sensor nodes to minimize the energy expenditure. The STBCS and SBCS

Figure 7.12: Energy consumption

are the second and third best approaches, which provide performance somewhat more comparable to our method than do the other methods. The main common feature of these three approaches is grouping sensor nodes and localizing sensor node measurements.

The means of grouping sensor nodes and gathering measurements from each group are the main difference among these methods. Since our approach groups the sensor nodes with reference to sparsity similarity, the number of active nodes is diminished. Moreover, it adjusts the number of measurements to the sparsity level, which removes the redundancy in data gathering. STBCS uses a clustering approach to group sensor nodes, but its grouping mechanism is not as accurate as that of our method. In addition, it does not consider adaptability to the number of measurements. Moreover, TBCS, BCS and OMP do not include localization of data gathering, which leads to greater exhaustion of energy.

7.7.3.4 Trade-off between energy and accuracy

We study the trade-off between network energy usage and data accuracy. To do so, we investigate energy consumption in terms of normalized network lifetime, while for accuracy we consider the error between the reconstructed data and real data.

As we can see in Figure 7.13, our approach provides approximately 40% network lifetime at a minimum error level, while with less accurate data it provides higher percentages of lifetime. Adapting both the number of measurements and the number of active sensor nodes to the signal sparsity level

Figure 7.13: Normalized network lifetime versus reconstruction error trade-Off

is the main contribution of our approach to prolong network lifetime. In addition, spatial-temporal correlation-based coalition formation provides sufficient similarity among sensor node measurements, which leads to less error in the recovered data.

7.8 Chapter summery

We have presented a distributed compressive sensing scheme that takes advantage of spatial temporal correlation of sensor node readings to achieve an energy efficient data collection. Spatial correlation among sensor nodes and sparsity distribution of signals over the network is used to group sensor nodes in the coalitions. The proposed coalition-formation method is represented by a block diagonal measurement matrix in which each diagonal entity corresponds to one of the coalitions. The localized spatial temporal correlation inside each coalition, builds an efficient measurement matrix which efficiently schedules sensor nodes and balances the data compression and transmission load over the coalition. Upon receiving compressed data at the base station, a two steps belief propagation based signal recovery is implemented to reconstruct the original signal. Having two steps signal recovery allows base station to find the common coefficient of original signal among coalitions and inside coalition. Therefore, it has enough data to recover data per sensor node level, which increases data reconstruction accuracy. Coalition-based data gathering is used to scale down transmission costs, while the number of measurements is

minimized by scheduling sensor nodes and adjusting their measurement rate. Our simulation on a real world data set proves our findings and shows more reduction on number of data measurements and compressibility as well as the ability to improve data reconstruction quality.

This chapter considers a network with stationary sensor nodes. However, there is a huge demand on wireless sensor networks with mobile nodes. In these networks, mobility of the nodes can change network topology. Proposed compressive sensing solution in this chapter can be extended in a way to cover the network topolgy changes.

Conclusion

In this chapter, we first study the summary of contributions and lessons learned in this thesis and then we give suggestions for future works.

8.1 Summary of contributions and lessons learned

Energy efficiency in wireless sensor networks is traditionally guaranteed by applying transmission cost reduction techniques on radio level. In other words, proposed data reduction based energy management techniques assume that radio communication is the most significant energy consumption parameter in the sensor networks. However, there are applications wherein the computational and sampling energy costs are comparable to the communication cost. The main lesson to learn here is that the energy efficiency on those applications can be ensured to some extent by reducing energy consumption on sensing and computation units.

This thesis proposed two data reduction techniques for wireless sensor networks to meet quality of service requirements and provide both energy efficiency and data quality. The proposed solutions focussed on two main data-reduction strategies: adaptive sampling and compressive sensing. Therefore, the thesis was thus divided into two parts.

The first part of this thesis elaborated on adaptive sampling techniques as a data reduction solution.

- **Chapter 2** presented a comprehensive review of the state of the art in the field of adaptive sampling. We presented a taxonomy of existing adaptive sampling approaches and compared them in terms of how well they meet the quality of service parameters. In this taxonomy, we classified adaptive sampling solutions into centralized and decentralized techniques. We then divided each of the classifications into temporal, spatial

and spatial-temporal subclasses. This chapter thus presented a thorough introduction to the adaptive sampling in wireless sensor networks. We compared the aforementioned approaches and their support for various quality of service parameters. Since in the centralized solution, sensor nodes have to transfer their all sampled data to the base station, proposed solutions can not best satisfy energy efficiency requirements. However, having a complete overview of the environmental conditions in the base station leads to more accurate data prediction. Furthermore, by considering the base station as a central decision making node for the whole network limits the scalability of the network. Our analysis shows that centralized solutions are suitable for small scale delay tolerant sensor networks. Simple sampling prediction in sensor node level can increase their scalability and energy efficiency.

On the other hand, the decentralized solutions localize the data sampling adjustment algorithms. In most of those solutions, base station and sensor nodes are equipped with the similar data prediction models. Therefore, the predicted data in the base station and sensor nodes are the same which minimizes data transmission necessity. Since the base station predicts sensor node data, it can report data to the end user with minimum delay. Furthermore, this localization improves the scalability and data quality of proposed solutions. For some scenarios, decentralized solutions suffer from extra communication overhead among sensor nodes.

Combining centralized and decentralized solutions helps hybrid methods to propose localized sampling scenarios and more accurate overview about environment condition in the base station. Therefore, energy efficiency and data quality requirements are mostly met. However, hybrid solutions suffer from complexity which should be improved.

- **Chapter 3** proposed an energy-efficient decentralized adaptive sampling method for wireless sensor nodes. Most of the researches in adaptive sampling area assume a small error threshold for sensor readings, which leads to tracking small changes in the data to meet the data-quality requirements of the application. The key lesson learned here is that, the environmental data like temperature, usually changes gradually, follows a normal distribution and falls into a specific range. Therefore, it is not needed to track all small changes. We proposed a new adaptive sampling solution which only tracks sensor node readings to see whether it falls inside a specific range. Basically, the sensor readings that fall within a given data range are not reported, however, those readings closer to

the boundaries of a specific range are reported as they have more values. Having this in mind, we take the advantage of spatial-temporal correlation among sensor nodes readings to provide energy-efficient scheduling for sensor nodes and to adjust the sampling frequency for each node. The main idea behind this approach was to use a carefully selected, dynamically changing subset of sensor nodes to sense and report data and to predict the values of non-sampler nodes in sleep mode. Furthermore, temporal correlation among sample node readings was used by all sensor nodes to tune their sampling frequency based on the environmental conditions. Through a simulation, we showed how our proposed method could effectively prolong network lifetime while satisfying data-quality metrics. Performance evaluation using a given dataset indicated reduction of energy consumption by the proposed approach, while event detection accuracy requested by the user/application was met.

- **Chapter 4** described a reward and punishment cooperative adaptive sampling method to satisfy both energy efficiency and data-quality requirements. Sensor nodes were allowed to decide whether to change their sampling frequency based on network-wide energy level and global data behavior pattern changes. The lesson learned in this chapter is related to the cooperation among sensor nodes to determine a proper sampling frequency for the entire network. This cooperation helped sensor nodes to adjust their sampling frequency based on the neighbor nodes status. We encouraged the sensor nodes to cooperate with each other to find the right balance between these two parameters by utilizing a reward and punishment system. When a sensor node detected frequent environmental changes, it increased its sampling rate, while other sensor nodes in its vicinity decreased their sampling rates to maintain the overall data quality of the whole network and keep energy expenditure low. Performance evaluation and comparison with other decentralized adaptive sampling approaches indicated that our solution minimizes energy consumption and increases data quality.

The second part of this thesis focussed on compressive sensing techniques.

- **Chapter 5** presented comprehensive review of data compression techniques. We classified existing compression methods into lossless and lossy methods and then sub-divided each of those classes into local and distributed methods. We used energy efficiency, scalability, compression ratio and data-quality parameters to study and evaluate the performance

of the solutions. Our studies revealed that lossless solutions provide better data quality which comes at the cost of more energy consumption. Dictionary and entropy based solutions represent a good compression efficiency, however, the existing spatial/temporal correlation in sensor networks are not utilized which limits their energy efficiency. On the other hand, distributed source coding based methods employ spatial/temporal correlation among sensor nodes and their readings and achieves higher compression and energy efficiency. Our studies on lossy compression techniques revealed that lossy data compression techniques show better compression ratios which results in better energy efficiency. The transform based compression and compressive sensing based techniques propose higher energy efficiency and data compression ratios. These solutions do not require any prior information about the sensor nodes readings and their correlation. However, they are not scalable for the large scale networks. Compressive sensing solutions represent simple encoding and complicated decoding procedures which best suits the sensor network characteristics. However, the complexity in the reconstruction procedure increases the latency parameter.

- **Chapter 6** represented the effect of transform bases on compressive sensing performance in terms of signal-to-noise ratio (SNR) and signal reconstruction error. Selecting an appropriate transform base for compressive sensing is crucial to sparse representation of signals, which can lead to fewer measurements and more accurate signal reconstruction. To the best of our knowledge, the selection of an appropriate transform basis has not been adequately studied, as most existing compressive sensing techniques have focussed on optimizing reconstruction algorithms. This chapter classifies different signals based on their characteristics and theorises the assignment of an appropriate transform basis for each specific type of signal. The comparison of simulation results using FT, HT and WT demonstrates that WT has the best performance when applying compressive sensing to this case study. Coiflet WT, Daubichi WT, Symlet WT, and Haar WT, the sub-families of WT, were also studied and analysed in terms of SNR and NRMSE.
- **Chapter 7** proposed a new data reduction technique using a novel distributed data compression technique. Most of the researches in data reduction and network coding area investigated on data sampling, encoding, compression, data transmission and decoding as individual research problem. This chapter utilized compressive sensing as a novel

data compression technique, which combined sensing, encoding, transmission, and decoding as a distributed data collection technique. Furthermore, it simplified data encoding process on resource limited sensor nodes level. The key lesson learned here was that novel research theories in data coding area had led to new data gathering solutions. However, those solutions would not be efficient and applicable as long as dealing with data sampling, encoding, and transmission and decoding techniques as separate problems. Compressive sensing mentioned in the second part of the thesis is a technique which introduces designing of new data collection techniques getting involved in all steps of data collection procedure from data sampling, encoding, transmission, network coding and finally data decoding. We proposed a spatial-temporal correlation based distributed compressive sensing solution for sensor networks. Proposed solution utilized sparsity distribution of signals to group sensor nodes into several coalitions and then implemented localized compressive sensing inside coalitions. This solution improved data-gathering performance in terms of both data accuracy and energy consumption. The approach curbs both data-transmission costs and number of measurements. Coalition-based data gathering cuts transmission cost, and the number of measurements was reduced by scheduling sensor nodes and adjusting their sampling frequency. A two-level belief-propagation-based reconstruction algorithm was proposed for acceptable data quality in terms of data accuracy. Our simulation showed that our approach enhances network performance by minimizing demand on energy and improving data accuracy.

8.2 Future research directions

This thesis has addressed certain problems related to the energy-efficient data collection in wireless sensor networks. However, issues remain for the future work. Some of these are described below.

- **Heterogeneous wireless sensor network:** This thesis focussed solely on homogenous wireless sensor networks, assuming that all sensor nodes are equipped with homogenous sensor nodes. However, many sensor network applications have sensor nodes with different computational power and sensor types. Our contribution can be extended to distribute data processing over sensor nodes based on their computational power. Furthermore, a data

fusion solution can be employed to combine different data types and increase information value at the base station.

- **Data aggregation:** Our efforts were solely directed to adaptive sampling and compressive sensing. However, further study is needed to evaluate the impact of our solutions when data aggregation is performed by each sensor node in the network. This is particularly important to reduce the amount of data transferred over the network.
- **Dynamic:** Our contributions in this thesis target networks with stationary sensor nodes. However, many applications employ mobile sensor and actuator nodes. The methods proposed in this thesis can be extended for use in mobile ad hoc networks, sensor and actuator networks and vehicular networks. The main challenge when applying our technique in these networks is related to the topology change. These networks experience dynamic topology changes during the network lifetime. As a future direction, our contribution can be extended to consider sensor nodes and link failures to capture dynamic topology changes.
- **Multi-hop network:** In the first part of this thesis, our method was applied on single-hop networks. However, the proposed solutions can be extended to include multi-hop networks as well. In this case, the main challenge is to combining adaptive sampling solutions with the data routing mechanism.
- **Uncertainty** The solution proposed here ignores any uncertainty that may be present in the sensor node observations. We did not consider this uncertainty as it was not a significant issue in our dataset. However, if we face a significant systematic error in sensor node reading, our solution will not be energy efficient anymore. In this case, sensor nodes require to sample more to degrease the impacts of uncertainty. Considering this uncertainty in proposed solution can be considered as a future work.

Bibliography

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] L. M. Oliveira and J. J. Rodrigues, "Wireless sensor networks: A survey on environmental monitoring," *JCM*, vol. 6, no. 2, pp. 143–151, 2011.
- [3] L. M. Borges, F. J. Velez, and A. S. Lebres, "Survey on the characterization and classification of wireless sensor network applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1860–1890, 2014.
- [4] K. J. Vitasek K., Manrodt K., *Minnesota Turns I-35W Bridge Tragedy into Triumph*. New York: Palgrave Macmillan, 2012, ch. 3, pp. 37–59.
- [5] G. Britain, *Railway Accident: Report on the Collapse of Penmanshiel Tunnel that Occurred on 17th March 1979 in the Scottish Region British Railways*. H.M. Stationery Office, 1983.
- [6] K. Fender, "Why europe's busiest railway collapsed at rastatt," 2017.
- [7] J. Chen, M. Díaz, L. Llopis, B. Rubio, and J. M. Troya, "A survey on quality of service support in wireless sensor and actor networks: Requirements and challenges in the context of critical infrastructure protection," *J. Network and Computer Applications*, vol. 34, no. 4, pp. 1225–1239, 2011.
- [8] D. Chen and P. K. Varshney, "Qos support in wireless sensor networks: A survey," in *International conference on wireless networks*, vol. 233, 2004, pp. 1–7.
- [9] F. Xia, "Qos challenges and opportunities in wireless sensor/actuator networks," *Sensors*, vol. 8, no. 2, pp. 1099–1110, 2008.
- [10] G. Verma and F. Dean, "Qos support in wireless sensor networks," *Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org*, vol. 5, no. 2, 2015.
- [11] J. Huang, H. Ogai, C. Shao, J. Zheng, I. Maruyama, S. Nagata, and H. Inujima, "On vibration signal analysis in bridge health monitoring system by using independent component analysis," in *Proc. SICE Annual Conf. 2010*, Aug. 2010, pp. 2122–2125.
- [12] M. A. Yigitel, O. D. Incel, and C. Ersoy, "Qos-aware mac protocols for wireless sensor networks: A survey," *Computer Networks*, vol. 55, no. 8, pp. 1982–2004, 2011.

- [13] R. C. Carrano, D. Passos, L. C. Magalhaes, and C. V. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 181–194, 2014.
- [14] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "Mac essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.
- [15] B. Yahya and J. Ben-Othman, "Towards a classification of energy aware mac protocols for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 9, no. 12, pp. 1572–1607, 2009.
- [16] J. B. Lim, B. Jang, and M. L. Sichitiu, "Mcas-mac: A multichannel asynchronous scheduled mac protocol for wireless sensor networks," *Computer Communications*, vol. 56, pp. 98–107, 2015.
- [17] J. Ansari, X. Zhang, and P. Mahonen, "Multi-radio medium access control protocol for wireless sensor networks," *International Journal of Sensor Networks*, vol. 8, no. 1, pp. 47–61, 2010.
- [18] Z. Liu and W. Wu, "A dynamic multi-radio multi-channel mac protocol for wireless sensor networks," in *Communication Software and Networks, 2010. ICCSN'10. Second International Conference on*. IEEE, 2010, pp. 105–109.
- [19] V. Namboodiri and A. Keshavarzian, "Alert: An adaptive low-latency event-driven mac protocol for wireless sensor networks," in *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*. IEEE, 2008, pp. 159–170.
- [20] M. C. Vuran and I. F. Akyildiz, "Spatial correlation-based collaborative medium access control in wireless sensor networks," *IEEE/ACM Transactions On Networking*, vol. 14, no. 2, pp. 316–329, 2006.
- [21] A. Klein, *Preamble-Based Medium Access in Wireless Sensor Networks*. INTECH Open Access Publisher, 2012.
- [22] F. Ahdi, W. Wang, V. Srinivasan, and K.-C. Chua, "Ips-mac: an informative preamble sampling mac protocol for wireless sensor networks," *Wireless Networks*, vol. 16, no. 5, pp. 1373–1387, 2010.
- [23] J. Inwhae, "A patterned preamble mac protocol for wireless sensor networks," *IEICE transactions on communications*, vol. 91, no. 2, pp. 658–661, 2008.
- [24] P. K. Pal and P. Chatterjee, "A survey on tdma-based mac protocols for wireless sensor network," *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 6, pp. 219–230, 2014.
- [25] D. Espes, X. Lagrange, and L. Suarez, "A cross-layer mac and routing protocol based on slotted aloha for wireless sensor networks," *annals of telecommunications-Annales des télécommunications*, vol. 70, no. 3-4, pp. 159–169, 2015.

- [26] M. D. Jovanovic and G. L. Djordjevic, "Ct-mac: Energy-efficient contention-based mac protocol for wireless sensor networks," *presentation in ICEST*, 2010.
- [27] R. Yadav, S. Varma, and N. Malaviya, "Optimized medium access control for wireless sensor network," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 2, pp. 334–338, 2008.
- [28] A. N. Alvi, S. H. Bouk, S. H. Ahmed, M. A. Yaqub, N. Javaid, and D. Kim, "Enhanced tdma based mac protocol for adaptive data control in wireless sensor networks," *Journal of communications and networks*, vol. 17, no. 3, pp. 247–255, 2015.
- [29] S. Ray, I. Demirkol, and W. Heinzelman, "Atma: advertisement-based tdma protocol for bursty traffic in wireless sensor networks," in *Global Telecommunications Conference (GLOBECOM 2010)*, 2010 IEEE. IEEE, 2010, pp. 1–5.
- [30] H. Gong, M. Liu, G. Chen, and X. Zhang, "A study on event-driven tdma protocol for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, p. 9, 2010.
- [31] J. Hong, I. Jang, H. Lee, S. Yang, and H. Yoon, "Mrmac: medium reservation mac protocol for reducing end-to-end delay and energy consumption in wireless sensor networks," *IEEE Communications Letters*, vol. 14, no. 7, pp. 614–616, 2010.
- [32] Y. Sun, O. Gurewitz, and D. B. Johnson, "Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 1–14.
- [33] P. Sthapit and J.-Y. Pyun, "Medium reservation based sensor mac protocol for low latency and high energy efficiency," *Telecommunication Systems*, pp. 1–9, 2013.
- [34] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Communications surveys & tutorials*, vol. 15, no. 2, pp. 551–591, 2013.
- [35] W. Guo and W. Zhang, "A survey on intelligent routing protocols in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 38, pp. 185–201, 2014.
- [36] O. Khan, F. G. Khan, B. Nazir, and U. Wazir, "Energy efficient routing protocols in wireless sensor networks: A survey," *International Journal of Computer Science and Information Security*, vol. 14, no. 6, p. 398, 2016.
- [37] W. R. H. J. Kulik and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," in *Wireless Networks*, 2002, pp. 169–185.
- [38] R. G. C. Intanagonwiwat and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *ACM MobiCom '00*, vol. 00, Boston, MA, 2000, pp. 56–67.

- [39] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *First Workshop on Sensor Networks and Applications (WSNA)*, , Atlanta, GA, 2002.
- [40] S. L. L. Z. F. Ye, A. Chen, "A scalable solution to minimum cost forwarding in large sensor networks," in *tenth International Conference on Computer Communications and Networks (ICCCN)*, 2001, pp. 304–309.
- [41] H. H. M. Chu and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heteroge- neous sensor networks," *The International Journal of High Performance Computing Applications*, vol. 16, no. 3, 2002.
- [42] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD Record*, 2002.
- [43] A. H. N. Sadagopan, B. Krishnamachari, "The acquire mechanism for efficient querying in sensor networks," in *First International Workshop on Sensor Network Protocol and Applications*, Anchorage, Alaska, 2003.
- [44] S. Servetto and G. Barrenechea, "Constrained random walks on random graphs: Routing algorithms for large scale wireless sensor networks," in *1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia, USA, 2002.
- [45] A. K. Yadav and P. Rana, "Position based routing schemes in wireless sensor networks: A review," in *Computational Intelligence & Communication Technology (CICT)*, 2016 *Second International Conference on.* IEEE, 2016, pp. 316–320.
- [46] K. Pavai, A. Sivagami, and D. Sridharan, "Study of routing protocols in wireless sensor networks," in *Advances in Computing, Control, & Telecommunication Technologies*, 2009. *ACT'09. International Conference on.* IEEE, 2009, pp. 522–525.
- [47] J. Son and T.-Y. Byun, "A routing scheme with limited flooding for wireless sensor networks," *International Journal of Future Generation Communication and Networking*, vol. 3, no. 3, pp. 33–40, 2010.
- [48] Z. Jin, Y. Jian-Ping, Z. Si-Wang, L. Ya-Ping, and L. Guang, "A survey on position-based routing algorithms in wireless sensor networks," *Algorithms*, vol. 2, no. 1, pp. 158–182, 2009.
- [49] I. Sim and J. Lee, "Routing protocol with scalability, energy efficiency and reliability in wsn," *Intelligent Automation & Soft Computing*, vol. 16, no. 4, pp. 567–577, 2010.
- [50] S. A. Madani, D. Weber, and S. Mahlknecht, "Position-based routing protocol for low power wireless sensor networks." *J. UCS*, vol. 16, no. 9, pp. 1215–1233, 2010.
- [51] A. M. Popescu, I. G. Tudorache, B. Peng, and A. H. Kemp, "Surveying position based routing protocols for wireless sensor and ad-hoc networks," *International Journal of Communication Networks and Information Security*, vol. 4, no. 1, p. 41, 2012.

- [52] H. Bagci and I. Korpeoglu, "Distributed and location-based multicast routing algorithms for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 697373, 2010.
- [53] J. You, Q. Han, D. Lieckfeldt, J. Salzmann, and D. Timmermann, "Virtual position based geographic routing for wireless sensor networks," *Computer Communications*, vol. 33, no. 11, pp. 1255–1265, 2010.
- [54] K. Bharathi, "A survey on cluster based routing protocol in wireless sensor network," *International Journal of Advances in Engineering & Technology*, vol. 9, no. 2, p. 186, 2016.
- [55] S. P. Singh and S. Sharma, "A survey on cluster based routing protocols in wireless sensor networks," *Procedia computer science*, vol. 45, pp. 687–695, 2015.
- [56] S. Manisekaran and S. Geetha, "Hierarchical energy efficient clustering algorithm for wireless sensor networks," in *International Conference on Computing*, 2012, p. 211.
- [57] J.-J. Liaw, C.-Y. Dai, and Y.-J. Wang, "The steady clustering scheme for heterogeneous wireless sensor networks," in *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on*. IEEE, 2009, pp. 336–341.
- [58] S. Tyagi and N. Kumar, "A systematic review on clustering and routing techniques based upon leach protocol for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 623–645, 2013.
- [59] D. Kumar, "Performance analysis of energy efficient clustering protocols for maximising lifetime of wireless sensor networks," *IET Wireless Sensor Systems*, vol. 4, no. 1, pp. 9–16, 2013.
- [60] K. Nawaz and A. P. Buchmann, "Acdmcp: an adaptive and completely distributed multi-hop clustering protocol for wireless sensor networks," *arXiv preprint arXiv:1009.0390*, 2010.
- [61] S. Manisekaran and R. Venkatesan, "An adaptive distributed power efficient clustering algorithm for wireless sensor networks," *American Journal of Scientific Research*, vol. 10, pp. 50–63, 2010.
- [62] A. Razaque, S. Mudigulam, K. Gavini, F. Amsaad, M. Abdulgader, and G. S. Krishna, "H-leach: Hybrid-low energy adaptive clustering hierarchy for wireless sensor networks," in *Long Island Systems, Applications and Technology Conference (LISAT), 2016 IEEE*. IEEE, 2016, pp. 1–4.
- [63] A. Forster and A. L. Murphy, "Clique: Role-free clustering with q-learning for wireless sensor networks," in *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on*. IEEE, 2009, pp. 441–449.
- [64] Q. A. Bakhtiar, K. Makki, and N. Pissinou, *Data reduction in low powered wireless sensor networks*. INTECH Open Access Publisher, 2012.

- [65] W. He, S.-H. Yang, L. Yang, and P. Li, "In-network data processing architecture for energy efficient wireless sensor networks," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE, 2015, pp. 299–304.
- [66] J. Chen, J. Ma, Y. Zhang, and X. Shi, "Ecg compression based on wavelet transform and golomb coding," *Electronics Letters*, vol. 42, no. 6, pp. 322–324, 2006.
- [67] G. Dhand and S. Tyagi, "Data aggregation techniques in wsn: Survey," *Procedia Computer Science*, vol. 92, pp. 378–384, 2016.
- [68] A. K. Talele, S. G. Patil, and N. B. Chopade, "A survey on data routing and aggregation techniques for wireless sensor networks," in *Pervasive Computing (ICPC), 2015 International Conference on*. IEEE, 2015, pp. 1–5.
- [69] D. S. Mantri, N. R. Prasad, and R. Prasad, "Bandwidth efficient cluster-based data aggregation for wireless sensor network," *Computers & Electrical Engineering*, vol. 41, pp. 256–264, 2015.
- [70] A. Sinha and D. K. Lobiyal, "Performance evaluation of data aggregation for cluster-based wireless sensor network," *Human-Centric Computing and Information Sciences*, vol. 3, no. 1, p. 13, 2013.
- [71] S. S. Ranjani, S. R. Krishnan, and C. Thangaraj, "Energy-efficient cluster based data aggregation for wireless sensor networks," in *Recent Advances in Computing and Software Systems (RACSS), 2012 International Conference on*. IEEE, 2012, pp. 174–179.
- [72] S. S. Sran, L. Kaur, G. Kaur, and S. K. Sidhu, "Energy aware chain based data aggregation scheme for wireless sensor network," in *Energy Systems and Applications, 2015 International Conference on*. IEEE, 2015, pp. 113–117.
- [73] S. Kaur and R. Gangwar, "A study of tree based data aggregation techniques for wsns," *International Journal of Database Theory and Application*, vol. 9, no. 1, pp. 109–118, 2016.
- [74] T.-W. Kuo, K. C.-J. Lin, and M.-J. Tsai, "On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: Np-completeness and approximation algorithms," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 3109–3121, 2016.
- [75] J. He, S. Ji, Y. Pan, and Y. Li, "Constructing load-balanced data aggregation trees in probabilistic wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1681–1690, 2014.
- [76] M. M. Gaber, U. Röhm, and K. Herink, "An analytical study of central and in-network data processing for wireless sensor networks," *Inf. Process. Lett.*, vol. 110, no. 2, pp. 62–70, 2009.
- [77] G. M. Dias, B. Bellalta, and S. Oechsner, "A survey about prediction-based data reduction in wireless sensor networks," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 58, 2016.

- [78] B. R. Stojkoska and K. Mahoski, "Comparison of different data prediction methods for wireless sensor networks," *CIIT, Bitola*, 2013.
- [79] J. Zhao, H. Liu, Z. Li, and W. Li, "Periodic data prediction algorithm in wireless sensor networks," in *China Conference on Wireless Sensor Networks*. Springer, 2012, pp. 695–701.
- [80] A. Deshpande and S. Madden, "Mauvedb: supporting model-based user views in database systems," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 73–84.
- [81] B. Kanagal and A. Deshpande, "Online filtering, smoothing and probabilistic modeling of streaming data," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE, 2008, pp. 1160–1169.
- [82] S. Bhandari, N. Bergmann, R. Jurdak, and B. Kusy, "Time series data analysis of wireless sensor network measurements of temperature," *Sensors*, vol. 17, no. 6, p. 1221, 2017.
- [83] J. Pardo, F. Zamora-Martínez, and P. Botella-Rocamora, "Online learning algorithm for time series forecasting suitable for low cost wireless sensor networks nodes," *Sensors*, vol. 15, no. 4, pp. 9277–9304, 2015.
- [84] Y.-A. Le Borgne, S. Santini, and G. Bontempi, "Adaptive model selection for time series prediction in wireless sensor networks," *Signal Processing*, vol. 87, no. 12, pp. 3010–3020, 2007.
- [85] N. Al-Hoqani and S.-H. Yang, "Adaptive sampling for wireless household water consumption monitoring," *Procedia Engineering*, vol. 119, pp. 1356–1365, 2015.
- [86] 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 Transactions on Instrumentation and Measurement*, vol. 59, no. 2, pp. 335–344, 2010.
- [87] S. Chatterjea and P. Havinga, "An adaptive and autonomous sensor sampling frequency control scheme for energy-efficient data acquisition in wireless sensor networks," in *International Conference on Distributed Computing in Sensor Systems*. Springer, 2008, pp. 60–78.
- [88] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 37–59, 2012.
- [89] M. A. Razzaque, C. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 1, p. 5, 2013.
- [90] T. Sheltami, M. Musaddiq, and E. Shakshuki, "Data compression techniques in wireless sensor networks," *Future Generation Computer Systems*, vol. 64, pp. 151–162, 2016.

- [91] Z. Yan-li, F. Xiao-ping, L. Shao-qiang, and X. Zhe-yuan, "Improved lzw algorithm of lossless data compression for wsn," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 4. IEEE, 2010, pp. 523–527.
- [92] C. Wang, S. R. Hussain, and E. Bertino, "Dictionary based secure provenance compression for wireless sensor networks," *IEEE transactions on parallel and distributed systems*, vol. 27, no. 2, pp. 405–418, 2016.
- [93] Y.-C. Wang, "Data compression techniques in wireless sensor networks," 2012.
- [94] E. Zimos, D. Toumpakaris, A. Munteanu, and N. Deligiannis, "Multiterminal source coding with copula regression for wireless sensor networks gathering diverse data," *IEEE Sensors Journal*, vol. 17, no. 1, pp. 139–150, 2017.
- [95] M. A. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors*, vol. 14, no. 2, pp. 2822–2859, 2014.
- [96] H. Zheng, F. Yang, X. Tian, X. Gan, X. Wang, and S. Xiao, "Data gathering with compressive sensing in wireless sensor networks: A random walk based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 35–44, Jan. 2015.
- [97] "Crossbow(imote2). datasheet of imote2,," 2011. [Online]. Available: <http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html>
- [98] "Crossbow(mica2).datasheet of mica2,," 2011. [Online]. Available: <http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html>
- [99] "Crossbow. datasheet of telosb,," 2011. [Online]. Available: http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf
- [100] "Freescale-semiconductor. datasheet of mma7260q,," 2010. [Online]. Available: http://www.freescale.com/files/sensors/doc/fact_sheet/MMA7260QFS.pdf
- [101] "Gems. datasheet of 2200 and 2600-series,," 2010. [Online]. Available: <http://www.gemssensors.com/Products/Pressure/Pressure-Tranducers/CVD//media/GemsNA/CatalogPages/2200-and-2600-series-cat.ashx>
- [102] "Intersil. datasheet of isl29002,," 2010. [Online]. Available: <http://www.intersil.com/data/fn/fn7465.pdf>
- [103] "Honeywell. datasheet of cp18 series,," 2012. [Online]. Available: http://content.honeywell.com/sensing/prodinfo/sds/installation/p80008_1.pdf
- [104] "Sensirion. datasheet of sht1x, 2010." 2010. [Online]. Available: <http://datasheet.octopart.com/SHT11-Sensirion-datasheet-5323722.pdf>
- [105] "Pepperl+fuchs. datasheet of luc-m10,," 2010. [Online]. Available: http://files.pepperl-fuchs.com/selector_files/navi/productInfo/edb/t47783_eng.pdf
- [106] "e2v. datasheet of mics-5521,," 2011. [Online]. Available: http://www.e2v.com/assets/media/files/sensors_datasheets/Metal_Oxide/mics-5521.pdf

- [107] "Turck. datasheet of fcs-gl1/2a4-ap8x-h1141,," 2011. [Online]. Available: http://www.nikhef.nl/pub/departments/mt/projects/zeus/vertex/Coolingsystem/PLC/Turck_flow.pdf
- [108] "Ge. datasheet of ge/telaire 6004,," 2011. [Online]. Available: http://www.veronics.com/products/infrared_gas_sensing/6004_CO2_Module.pdf
- [109] Y. Jon, "Adaptive sampling in wireless sensor networks for air monitoring system," 2016.
- [110] B. Shahrasbi, A. Talari, and N. Rahnavard, "Tc-csbp: Compressive sensing for time-correlated data based on belief propagation," in *Proc. 45th Annual Conf. Information Sciences and Systems*, Mar. 2011, pp. 1–6.
- [111] M. Leinonen, M. Codreanu, and M. Juntti, "Sequential compressed sensing with progressive signal reconstruction in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1622–1635, 2015.
- [112] I. L. Data., 2004. [Online]. Available: <http://db.csail.mit.edu/labdata/labdata.html>
- [113] (2008) Sensorscope luce deployment at ecole polytechnique federale de lausanne,. [Online]. Available: <http://lcav.epfl.ch/op/edit/sensorscope-en>.
- [114] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.
- [115] C. Alippi, G. Anastasi, C. Galperti, F. Mancini, and M. Roveri, "Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. IEEE, 2007, pp. 1–6.
- [116] C. Alippi and M. Roveri, "An adaptive cusum-based test for signal change detection," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 2006, pp. 4 pp.–.
- [117] A. Jain and E. Y. Chang, "Adaptive sampling for sensor networks," in *Proceedings of the 1st International Workshop on Data Management for Sensor Networks: In Conjunction with VLDB 2004*, ser. DMSN '04. New York, NY, USA: ACM, 2004, pp. 10–16. [Online]. Available: <http://doi.acm.org/10.1145/1052199.1052202>
- [118] J. Zhou and D. De Roure, "Floodnet: Coupling adaptive sampling with energy aware routing in a flood warning system," *Journal of Computer Science and Technology*, vol. 22, no. 1, pp. 121–130, 2007.
- [119] R. Willett, A. Martin, and R. Nowak, "Backcasting: adaptive sampling for sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 124–133.
- [120] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. D. Moor, D. S. Bernstein, and A. Ridley, "What is the ensemble Kalman filter and how well does it work?" in *Proc. American Control Conf*, Jun. 2006, pp. 6 pp.–.

- [121] M. Vahabi, M. Rasid, R. Abdullah, and M. Ghazvini, "Adaptive data collection algorithm for wireless sensor networks," *International Journal of Computer Science and Network Security*, vol. 8, no. 6, pp. 125–132, 2008.
- [122] S. Lin, B. Arai, D. Gunopulos, and G. Das, "Region sampling: Continuous adaptive sampling on sensor networks," in *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008, pp. 794–803.
- [123] M. Mathew and N. Weng, "Quality of information and energy efficiency optimization for sensor networks via adaptive sensing and transmitting," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 341–348, 2014.
- [124] M. Perillo, Z. Ignjatovic, and W. Heinzelman, "An energy conservation method for wireless sensor networks employing a blue noise spatial sampling technique," in *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*. IEEE, 2004, pp. 116–123.
- [125] T. Dang, N. Bulusu, W.-c. Feng, S. Frolov, and A. Baptista, "Adaptive sampling in the columbia river observation network," in *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 2007, pp. 429–430.
- [126] R. Van Der Merwe, E. A. Wan, S. Julier *et al.*, "Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," in *Proceedings of the AIAA Guidance, Navigation & Control Conference*, 2004, pp. 16–19.
- [127] M. Rahimi, M. Hansen, W. J. Kaiser, G. S. Sukhatme, and D. Estrin, "Adaptive sampling for environmental field estimation using robotic sensors," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3692–3698.
- [128] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 588–599.
- [129] B. Gedik, L. Liu, and S. Y. Philip, "Asap: an adaptive sampling approach to data collection in sensor networks," *IEEE Transactions on Parallel and distributed systems*, vol. 18, no. 12, pp. 1766–1783, 2007.
- [130] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes," in *Proceedings of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society, 2008, pp. 109–120.
- [131] A. Kose and E. Masazade, "Adaptive sampling with sensor selection for target tracking in wireless sensor networks," in *2014 48th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2014, pp. 909–913.

- [132] H. Wu and Q. Luo, "Supporting adaptive sampling in wireless sensor networks," in *2007 IEEE Wireless Communications and Networking Conference*. IEEE, 2007, pp. 3442–3447.
- [133] P. Padhy, R. K. Dash, K. Martinez, and N. R. Jennings, "A utility-based sensing and communication model for a glacial sensor network," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM, 2006, pp. 1353–1360.
- [134] Y. W. Law, S. Chatterjea, J. Jin, T. Hanselmann, and M. Palaniswami, "Energy-efficient data acquisition by adaptive sampling for wireless sensor networks," in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*. ACM, 2009, pp. 1146–1151.
- [135] A. Jain, E. Y. Chang, and Y.-F. Wang, "Adaptive stream resource management using kalman filters," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 2004, pp. 11–22.
- [136] J. Kho, A. Rogers, and N. R. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, p. 19, 2009.
- [137] C. Salim, A. Makhoul, R. Darazi, and R. Couturier, "Adaptive sampling algorithms with local emergency detection for energy saving in wireless body sensor networks," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 2016, pp. 745–749.
- [138] S. Santini and K. Romer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," in *Proceedings of the 3rd international conference on networked sensing systems (INSS 2006)*, 2006, pp. 29–36.
- [139] Z. Quan, W. J. Kaiser, and A. H. Sayed, "A spatial sampling scheme based on innovations diffusion in sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 323–330.
- [140] Z. Quan and A. H. Sayed, "Innovations-based sampling over spatially-correlated sensors," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 3. IEEE, 2007, pp. III–509.
- [141] K. Sha and W. Shi, "Consistency-driven data quality management of networked sensor systems," *Journal of Parallel and Distributed Computing*, vol. 68, no. 9, pp. 1207–1221, 2008.
- [142] E. K. Lee, H. Viswanathan, and D. Pompili, "Distributed data-centric adaptive sampling for cyber-physical systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 9, no. 4, p. 21, 2015.
- [143] M. Zhang, W. Cai, L. Zhou, and J. Liu, "Energy-effective frequency-based adaptive sampling algorithm for clustered wireless sensor network," in *Computer Engineering and Networking*. Springer, 2014, pp. 107–114.

- [144] G. Mainland, D. C. Parkes, and M. Welsh, "Decentralized, adaptive resource allocation for sensor networks," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 315–328.
- [145] V. L. Nguyen, "Adaptive sampling for spatial prediction in wireless sensor networks," Ph.D. dissertation, 2014.
- [146] 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*, vol. 16, no. 4, p. 448, 2016.
- [147] M. N. Halgamuge, M. Zukerman, K. Ramamohanarao, and H. L. Vu, "An estimation of sensor energy consumption," *Progress In Electromagnetics Research B*, 2009.
- [148] R. A. Brockwell, P. J.; Davis, *Time Series: Theory and Methods*. Springer, 2009.
- [149] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [150] R. A. Chumacero, "Estimating arma models efficiently," 2001.
- [151] "Mica2dot datasheet." 2004. [Online]. Available: <http://www.cmt-gmbh.de/Mica2dot.pdf>
- [152] G. Aiello, G. La Scalia, and R. Micale, "Simulation analysis of cold chain performance based on time–temperature data," *Production Planning & Control*, vol. 23, no. 6, pp. 468–476, 2012.
- [153] M. Hifi, M. Michrafy, and A. Sbihi, "Heuristic algorithms for the multiple-choice multidimensional knapsack problem," *JORS*, vol. 55, no. 12, pp. 1323–1332, 2004.
- [154] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104–122, 2014.
- [155] T. M. Cover, "Elements of information theory thomas m. cover, joy a. thomas copyright© 1991 john wiley & sons, inc. print isbn 0-471-06259-6 online isbn 0-471-20061-1," 1991.
- [156] S.-W. Seong, "Dictionary-based code compression techniques using bit-masks for embedded systems," Ph.D. dissertation, University of Florida, 2006.
- [157] J.-N. Hwang, *Multimedia networking: from theory to practice*. Cambridge University Press, 2009.
- [158] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *IEEE communications letters*, vol. 12, no. 6, 2008.
- [159] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Transactions on Information Systems (TOIS)*, vol. 16, no. 3, pp. 256–294, 1998.

- [160] C. Tharini and P. V. Ranjan, "Design of modified adaptive huffman data compression algorithm for wireless sensor network," *Journal of Computer Science*, vol. 5, no. 6, p. 466, 2009.
- [161] Y. Liang and W. Peng, "Minimizing energy consumptions in wireless sensor networks via two-modal transmission," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 12–18, 2010.
- [162] J. G. Kolo, S. A. Shanmugam, D. W. G. Lim, L.-M. Ang, and K. P. Seng, "An adaptive lossless data compression scheme for wireless sensor networks," *Journal of Sensors*, vol. 2012, 2012.
- [163] I. M. A. D. Suarjaya, "A new algorithm for data compression optimization," *CoRR*, vol. abs/1209.1045, 2012. [Online]. Available: <http://arxiv.org/abs/1209.1045>
- [164] E. P. Capo-Chichi, H. Guyennet, and J. M. Friedt, "K-rle: A new data compression algorithm for wireless sensor network," in *Proc. Third Int. Conf. Sensor Technologies and Applications*, Jun. 2009, pp. 502–507.
- [165] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51–60, 2002.
- [166] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE signal processing magazine*, vol. 21, no. 5, pp. 80–94, 2004.
- [167] Z. Tang, I. A. Glover, A. N. Evans, and J. He, "An energy-efficient adaptive DSC scheme for wireless sensor networks," *Signal Processing*, vol. 87, no. 12, pp. 2896–2910, 2007.
- [168] F. Oldewurtel, J. Riihijarvi, and P. Mahonen, "Efficiency of distributed compression and its dependence on sensor node deployments," in *Vehicular Technology Conference (VTC 2010-Spring)*, 2010 IEEE 71st. IEEE, 2010, pp. 1–5.
- [169] Y. Yang, S. Guo, and Y. Yang, "Distributed optimal source coding rate allocation for data aggregation in wireless sensor networks," in *Parallel and Distributed Systems (ICPADS)*, 2016 IEEE 22nd International Conference on. IEEE, 2016, pp. 32–39.
- [170] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [171] S.-T. Cheng, J.-S. Shih, C.-L. Chou, G.-J. Horng, and C.-H. Wang, "Hierarchical distributed source coding scheme and optimal transmission scheduling for wireless sensor networks," *Wireless personal communications*, vol. 70, no. 2, pp. 847–868, 2013.
- [172] C. C. Cutler, "Differential quantization of communication signals," Jul. 29 1952, uS Patent 2,605,361.

- [173] F. Marcelloni and M. Vecchio, "Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization," *Information Sciences*, vol. 180, no. 10, pp. 1924–1941, 2010.
- [174] D. I. Sacaleanu, R. Stoian, D. M. Ofrim, and N. Deligiannis, "Compression scheme for increasing the lifetime of wireless intelligent sensor networks," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 709–713.
- [175] F. Marcelloni and M. Vecchio, "Exploiting multi-objective evolutionary algorithms for designing energy-efficient solutions to data compression and node localization in wireless sensor networks," in *Evolutionary Based Solutions for Green Computing*. Springer, 2013, pp. 227–255.
- [176] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [177] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, "Lightweight temporal compression of microclimate datasets," *Center for Embedded Network Sensing*, 2004.
- [178] K. Kavitha, D. Sharma, R. Surana, and R. Gururaj, "Induced redundancy based lossy data compression algorithm," *International Journal of Computer Applications*, vol. 62, no. 16, 2013.
- [179] A. Oka and L. Lampe, "Incremental distributed identification of markov random field models in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2396–2405, 2009.
- [180] M. F. Duarte, G. Shen, A. Ortega, and R. G. Baraniuk, "Signal compression in wireless sensor networks," *Phil. Trans. R. Soc. A*, vol. 370, no. 1958, pp. 118–135, 2012.
- [181] A. Amar, A. Leshem, and M. Gastpar, "Recursive implementation of the distributed karhunen-loève transform," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5320–5330, 2010.
- [182] A. Bertrand and M. Moonen, "Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed pca," *Signal Processing*, vol. 104, pp. 120–135, 2014.
- [183] A. Ciancio and A. Ortega, "A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 4. IEEE, 2005, pp. iv–825.
- [184] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, 2006, pp. 309–316.

- [185] T. Dang, N. Bulusu, and W.-c. Feng, "Robust data compression for irregular wireless sensor networks using logical mapping," *ISRN Sensor Networks*, vol. 2013, 2013.
- [186] G. Shen, "Lifting transforms on graphs: Theory and applications," Ph.D. dissertation, University of Southern California, 2010.
- [187] S. Yildirim, A. T. Cemgil, and S. S. Singh, "An online expectation-maximisation algorithm for nonnegative matrix factorisation models," *CoRR*, vol. abs/1401.2490, 2014. [Online]. Available: <http://arxiv.org/abs/1401.2490>
- [188] E. H. Aoki, A. Bagchi, P. Mandal, and Y. Boers, "A theoretical look at information-driven sensor management criteria," in *Proc. 14th Int. Conf. Information Fusion*, Jul. 2011, pp. 1–8.
- [189] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, 2008.
- [190] Z. Zhang, T. P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 1, pp. 221–224, Jan. 2013.
- [191] F. A. Aderohunmu, G. Paci, L. Benini, J. D. Deng, and D. Brunelli, "Swiftnet: A data acquisition protocol for fast-reactive monitoring applications," in *Proc. 8th IEEE Int. Symp. Industrial Embedded Systems (SIES)*, Jun. 2013, pp. 93–96.
- [192] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Proc. 5th Int. Conf. Information Processing in Sensor Networks*, Apr. 2006, pp. 134–142.
- [193] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Efficient measurement generation and pervasive sparsity for compressive data gathering," *IEEE Transactions on Wireless Communications*, vol. 9, no. 12, pp. 3728–3738, Dec. 2010.
- [194] H. Hu and Z. Yang, "Spatial correlation-based distributed compressed sensing in wireless sensor networks," in *Proc. 6th Int. Conf. Wireless Communications Networking and Mobile Computing (WiCOM)*, Sep. 2010, pp. 1–4.
- [195] J. Meng, H. Li, and Z. Han, "Sparse event detection in wireless sensor networks using compressive sensing," in *Proc. 43rd Annual Conf. Information Sciences and Systems*, Mar. 2009, pp. 181–185.
- [196] X. Yang, X. Tao, E. Dutkiewicz, X. Huang, Y. J. Guo, and Q. Cui, "Energy-efficient distributed data storage for wireless sensor networks based on compressed sensing and network coding," *IEEE Transactions on Wireless Communications*, vol. 12, no. 10, pp. 5087–5099, Oct. 2013.
- [197] L. Cheng, J. Niu, L. Kong, C. Luo, Y. Gu, W. He, and S. K. Das, "Compressive sensing based data quality improvement for crowd-sensing applications," *Journal of Network and Computer Applications*, vol. 77, pp. 123–134, 2017.

- [198] S. Chen, C. Zhao, M. Wu, Z. Sun, and J. Jin, "Clustered spatio-temporal compression design for wireless sensor networks," in *Computer Communication and Networks (ICCCN), 2015 24th International Conference on*. IEEE, 2015, pp. 1–6.
- [199] L. Kong, M. Xia, X.-Y. Liu, G. Chen, Y. Gu, M.-Y. Wu, and X. Liu, "Data loss and reconstruction in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2818–2828, 2014.
- [200] R. Baraniuk, M. A. Davenport, M. F. Duarte, C. Hegde *et al.*, "An introduction to compressive sensing," *Connexions e-textbook*, 2011.
- [201] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [202] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14–20, 2008.
- [203] L. Xu, "Implementation of new strategies in multidimensional compressive imaging," Ph.D. dissertation, Rice University, 2014.
- [204] T. Wescott, "Sampling: what nyquist didn't say, and what to do about it," Wescott Design Services, Jun. 2016.
- [205] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4053–4085, Sep. 2011.
- [206] J. Haupt, L. Applebaum, and R. Nowak, "On the restricted isometry of deterministically subsampled Fourier matrices," in *Proc. 44th Annual Conf. Information Sciences and Systems (CISS)*, Mar. 2010, pp. 1–6.
- [207] D. S. Kalogerias and A. P. Petropulu, "Rip bounds for naively subsampled scrambled Fourier sensing matrices," in *Proc. 48th Annual Conf. Information Sciences and Systems (CISS)*, Mar. 2014, pp. 1–6.
- [208] S. Haykin and B. Van Veen, *Signals and systems*. John Wiley & Sons, 2007.
- [209] G. Li, L. Chang, and S. Li, *Signals and Systems: Fundamentals*. Walter de Gruyter GmbH & Co KG, 2015.
- [210] A. C. Gilbert, M. J. Strauss, and J. A. Tropp, "A tutorial on fast fourier sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 57–66, 2008.
- [211] T. Canli, A. Gupta, and A. Khokhar, "Power efficient algorithms for computing fast Fourier transform over wireless sensor networks," in *Proc. IEEE Int. Conf. Computer Systems and Applications*, Mar. 2006, pp. 549–556.
- [212] D. Guo, X. Qu, M. Xiao, and Y. Yao, "Comparative analysis on transform and reconstruction of compressed sensing in sensor networks," in *Proc. WRI Int. Conf. Communications and Mobile Computing*, vol. 1, Jan. 2009, pp. 441–445.
- [213] P. M. Bentley and J. T. E. McDonnell, "Wavelet transforms: an introduction," *Electronics Communication Engineering Journal*, vol. 6, no. 4, pp. 175–186, Aug. 1994.

- [214] L. Chun-Lin, "A tutorial of the wavelet transform," *NTUET, Taiwan*, 2010.
- [215] J. Zheng, Q. Wang, H. Ogai, C. Shao, and J. Huang, "Bridge diagnosis system by using nonlinear independent component analysis," in *Proc. SICE Annual Conf. 2010*, Aug. 2010, pp. 2118–2121.
- [216] A. Al-Radaideh, A. R. Al-Ali, S. Bheiry, and S. Alawnah, "A wireless sensor network monitoring system for highway bridges," in *Proc. Int. Conf. Electrical and Information Technologies (ICEIT)*, Mar. 2015, pp. 119–124.
- [217] T. Yu-liang, L. Yu, H. Lian-fen, G. Jian, and L. Ying, "Wireless sensor network for on-line structural health monitoring," in *Proc. 7th Int. Conf. Computer Science Education (ICCSE)*, Jul. 2012, pp. 386–389.
- [218] T. Harms, S. Sedigh, and F. Bastianini, "Structural health monitoring of bridges using wireless sensor networks," *IEEE Instrumentation Measurement Magazine*, vol. 13, no. 6, pp. 14–18, Dec. 2010.
- [219] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Commun. ACM*, vol. 53, no. 12, pp. 93–100, Dec. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1859204.1859229>
- [220] M. Balouchestani, K. Raahemifar, and S. Krishnan, "Robust wireless sensor networks with compressed sensing theory," *Networked Digital Technologies*, pp. 608–619, 2012.
- [221] J. F. Villaverde, W.-Y. Chung, and S.-L. Chen, "Compressive sensing algorithm for wireless sensor network power management," *International Journal of Computer and Electrical Engineering*, vol. 7, no. 3, p. 1, 2015.
- [222] Y. Quer, R. Masiero, M. Rossi, and M. Zorzi, "Sensing, compression and recovery for wireless sensor networks: monitoring framework design," *Wireless Communications, IEEE Transactions*, vol. 11, pp. 3447–3461, 2012.
- [223] C. T. Chou, R. Rana, and W. Hu, "Energy efficient information collection in wireless sensor networks using adaptive compressive sensing," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*. IEEE, 2009, pp. 443–450.
- [224] W. Chen and I. J. Wassell, "Energy-efficient signal acquisition in wireless sensor networks: a compressive sensing framework," *IET wireless sensor systems*, vol. 2, no. 1, pp. 1–8, 2012.
- [225] R. Middy, N. Chakravarty, and M. K. Naskar, "Compressive sensing in wireless sensor networks—a survey," *IETE Technical Review*, pp. 1–13, 2016.
- [226] C. Agrawal and D. Ghosh, "Distributed compressive data gathering in wireless sensor networks," in *Proc. IEEE 11th Int. Conf. Signal Processing*, vol. 3, Oct. 2012, pp. 2110–2115.
- [227] M. Laifenfeld and I. Bilik, "Distributed compressive sensing and communications in wireless sensor networks," in *Proc. IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, Nov. 2012, pp. 1–5.

- [228] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.
- [229] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 269–280, 2010.
- [230] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," *arXiv preprint arXiv:0901.3403*, 2009.
- [231] D. Sundman, S. Chatterjee, and M. Skoglund, "Methods for distributed compressed sensing," *Journal of Sensor and Actuator Networks*, vol. 3, no. 1, pp. 1–25, 2013.
- [232] Z. Tan, P. Yang, and A. Nehorai, "Joint sparse recovery method for compressed sensing with structured dictionary mismatches," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 4997–5008, 2014.
- [233] S. Khanna and C. R. Murthy, "Decentralized joint-sparse signal recovery: A sparse bayesian learning approach," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 1, pp. 29–45, Mar. 2017.
- [234] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 912–926, Sep. 2011.
- [235] G. Tzagkarakis, J.-L. Starck, and P. Tsakalides, "Joint sparse signal ensemble reconstruction in a wsn using decentralized bayesian matching pursuit," in *Signal Processing Conference, 2011 19th European*. IEEE, 2011, pp. 338–342.
- [236] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 340–354, 2013.
- [237] V. Cevher, P. Indyk, L. Carin, and R. G. Baraniuk, "Sparse signal recovery and acquisition with graphical models," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 92–103, Nov. 2010.
- [238] T. Gannon, *Vertex operator algebras*, P. C. to Mathematics, Ed. Princeton University Press., 2008.
- [239] C. Caione, D. Brunelli, and L. Benini, "Compressive sensing optimization over zigbee networks," in *IEEE Fifth International Symposium on Industrial Embedded Systems, SIES 2010, University of Trento, Italy, July 7-9, 2010*. IEEE, 2010, pp. 36–44.
- [240] M. Nguyen and K. Teague, "Energy-efficient data collection in clustered wireless sensor networks employing distributed DCT," *CoRR*, vol. abs/1701.02606, 2017. [Online]. Available: <http://arxiv.org/abs/1701.02606>

- [241] M. Mahmudimanesh, "Efficient spatio-temporal sampling in wireless sensor networks based on compressive sampling," Ph.D. dissertation, Darmstadt University of Technology, Germany, 2015. [Online]. Available: <http://tuprints.ulb.tu-darmstadt.de/4944/>
- [242] R. Masiero, G. Quer, M. Rossi, and M. Zorzi, "A bayesian analysis of compressive sensing data recovery in wireless sensor networks," in *Ultra Modern Telecommunications & Workshops, 2009. ICUMT'09. International Conference on.* IEEE, 2009, pp. 1–6.
- [243] X. Wu, Y. Xiong, M. Li, and W. Huang, "Distributed spatial-temporal compressive data gathering for large-scale wsns," in *Proc. Communications and IT Applications Conf 2013 Computing (ComComAp)*, Apr. 2013, pp. 105–110.
- [244] S. Ji, L. Huang, J. Wang, and Q. Liu, "A reconstruction method of compressive sensing data recovery in wireless sensor networks for shm," 2013.
- [245] S. Hwang, R. Ran, J. Yang, and D. K. Kim, "Multivariate bayesian compressive sensing in wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 7, pp. 2196–2206, 2016.

About the author

Alireza Masoum obtained his B.Sc. degree in Computer Engineering from the Ferdowsi University of Mashad, Iran, in 2003. Later, he obtained his M.Sc. degree in computer architecture from the Sharif University of Technology, Iran, in 2005. In November 2009, he joined the Pervasive Systems group at the University of Twente to pursue his Ph.D. degree in the field of data reduction in wireless sensor networks. During this time, he worked on the FREE and GEN-ESI research projects.

List of publications in which he participated in reverse chronological order:

- 1) A. Masoum, N. Meratnia and P.J.M. Havinga, "Coalition formation based compressive sensing in wireless sensor networks"; submitted to Sensors Journal, <http://www.mdpi.com/journal/sensors/sections/sensornetworks>
- 2) A. Masoum, N. Meratnia and P.J.M. Havinga, "Compressive sensing based data collection in wireless sensor networks," IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), South Korea, 2017(2017, Page(s):442 - 447), DOI:10.1109/MFI.2017.8170360
- 3) A. Masoum, N. Meratnia and P.J.M. Havinga, "A distributed compressive sensing technique for data gathering in Wireless Sensor Networks," In The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks, EUSPN 2013 (pp. 207-216). (Procedia Computer Science; Vol. 21). Amsterdam: ELSEVIER. DOI: 10.1016/j.procs.2013.09.028
- 4) A. Masoum, N. Meratnia and P.J.M. Havinga, "An energy-efficient adaptive sampling scheme for wireless sensor networks," In Eighth IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing, IEEE ISSNIP 2013 (pp. 231-236). USA: IEEE Computer Society. DOI: 10.1109/ISSNIP.2013.6529794

- 5) A. Masoum, N. Meratnia and P.J.M. Havinga, "Analysis of the impact of data correlation on adaptive sampling in Wireless Sensor Networks," In Ninth International Conference on Networked Sensing Systems, INSS 2012 (pp. 1-2). USA: IEEE. DOI: 10.1109/INSS.2012.6240565
- 6) A. Masoum, N. Meratnia and P.J.M. Havinga, "A Decentralized Quality Aware Adaptive Sampling Strategy in Wireless Sensor Networks," In 9th IEEE International Conference on Ubiquitous Intelligence and Computing, UIC 2012 (pp. 298-305). USA: IEEE Computer Society. DOI: 10.1109/UIC-ATC.2012.156
- 7) A. Masoum, N. Meratnia, A. Dilo, Z. Taghikhaki and P.J.M. Havinga, "Cross-layer analyses of QoS parameters in wireless sensor networks," In Proceedings of the First International Conference on Computer Science and Information Technology, CCSIT (pp. 595-605). (Communications in Computer and Information Science; Vol. 132). Berlin: Springer Verlag. DOI: 10.1007/978-3-642-17878-8-60
- 8) A. Masoum, N. Meratnia, Z. Taghikhaki and P.J.M. Havinga, "Reward and Punishment based Cooperative Adaptive Sampling in Wireless Sensor Networks," In Proceedings of the 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) (pp. 145-150). USA: IEEE Computer Society. DOI: 10.1109/ISSNIP.2010.5706787

ISBN : 978-90-365-4564-8