Ministério da
Ciência e Tecnologia

INPE-15669-TDI/1444

DATA SUPRESSION IN SENSOR NETWORKS: IMPROVISING THE QUALITY OF ESTIMATES AND THE ROBUSTNESS TO ABERRANT READINGS

Ilka Afonso Reis

Tese de Doutorado do Curso de Pós-Graduação em Sensoriamento Remoto, orientada pelo Dr. Gilberto Câmara, aprovada em 18 de dezembro de 2008

Registro do documento original:
<http://urlib.net/sid.inpe.br/mtc-m18@80/2008/11.19.20.24>

INPE
São José dos Campos
2009

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3945-6911/6923
Fax: (012) 3945-6919
E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO:
Presidente:
Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)
Membros:
Drª Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação
Dr. Haroldo Fraga de Campos Velho - Centro de Tecnologias Especiais (CTE)
Drª Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)
Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)
Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)
Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)
BIBLIOTECA DIGITAL:
Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)
Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)
Jefferson Andrade Ancelmo - Serviço de Informação e Documentação (SID)
Simone A. Del-Ducca Barbedo - Serviço de Informação e Documentação (SID)
REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:
Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)
Marilúcia Santos Melo Cid - Serviço de Informação e Documentação (SID)
Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)
EDITORAÇÃO ELETRÔNICA:
Viveca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)

Ministério da
Ciência e Tecnologia

INPE-15669-TDI/1444

DATA SUPRESSION IN SENSOR NETWORKS: IMPROVISING THE QUALITY OF ESTIMATES AND THE ROBUSTNESS TO ABERRANT READINGS

Ilka Afonso Reis

Tese de Doutorado do Curso de Pós-Graduação em Sensoriamento Remoto, orientada pelo Dr. Gilberto Câmara, aprovada em 18 de dezembro de 2008

Registro do documento original:
<http://urlib.net/sid.inpe.br/mtc-m18@80/2008/11.19.20.24>

INPE
São José dos Campos
2009

Dados Internacionais de Catalogação na Publicação (CIP)

R375d Reis, Ilka Afonso.
Data supression in Sensor networks: Improvising the quality of estimates and the robustness to aberrant readings / Ilka Afonso Reis. – São José dos Campos: INPE, 2009.
168p. ; (INPE-15669-TDI/1444)

Dissertação (Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2008.

1. Sensor network. 2. Data suppression. 3. Data collection. 4. Outliers. 5. Environmental monitoring. I.Título.

CDU 528.83:528.85

Copyright © 2009 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecánico, fotográfico, microfílmico, reprográfico ou outros, sem a permissão escrita da Editora, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2009 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, eletronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de Doutor(a) em

Sensoriamento Remoto

Dr. Antonio Miguel Vieira Monteiro

Presidente / Orientador(a) / INPE / SJCampos - SP

Dr. Gilberto Câmara

Orientador(a) / INPE / SJCampos - SP

Dr. Camilo Daleles Rennó

Membro da Banca / INPE / SJCampos - SP

Dr. Renato Martins Assunção

Convidado(a) / UFMG / Belo Horizonte - MG

Dr. Alejandro Cesar Frery Orgambide

Convidado(a) / UFAL / Maceió - AL

Aluno (a): Ilka Afonso Reis

São José dos Campos, 18 de dezembro de 2008

"An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem."

J. W. Tukey (1915★ – 2000✟)

Para meus queridos pais, Dálvio♥ e Oraida, que acreditaram ser a educação o melhor presente que poderiam me oferecer.

♥ Eterna saudade

AGRADECIMENTOS

Por mais solitário que possa ter parecido realizar este trabalho, sei que tudo teria sido muito mais difícil (ou mesmo impossível) sem o apoio de várias pessoas. Neste espaço, gostaria de sintetizar meus sinceros agradecimentos a algumas delas:

A meu orientador, Dr. Gilberto Câmara, por ter acreditado no meu potencial para o desenvolvimento deste projeto e por ter contribuído para o meu crescimento no mundo da Ciência (e dos vinhos).

A meus colaboradores, Dr. Antônio Miguel Monteiro e Prof. Renato Assunção, pelo grande apoio recebido durante a elaboração deste trabalho.

A meu esposo Agostinho, pelo companheirismo, pelo apoio incondicional a este projeto, pela paciência nos meus momentos impacientes e por estar sempre presente quando mais dele precisei.

A meus pais, Dálvio e Oraida, e às minhas irmãs, Edna e Tânia, por formarem a família na qual sempre pude encontrar amor, apoio e fé.

A meus colegas da PG-SERE, em especial, Giovana, Elienê, Daniela, Vanessa, Wilson, Murilo, Eduardo Araújo e Roberto, e às minhas colegas da PG-CAP, Evaldinólia, Karla, Missae e Olga, pela amizade, suporte e pelos bons momentos vividos.

Aos funcionários da PG-SERE, em especial à Etel, e da OBT, em especial, à Dra. Silvana Amaral, pelo apoio recebido durante o tempo em que estive no INPE.

Aos membros da banca examinadora, Prof. Alejandro Frery e Dr. Camilo Rennó, pelo cuidado que tiveram ao ler este documento e pelas sugestões oferecidas.

Ao Departamento de Estatística da UFMG, por ter prescindido da minha colaboração durante este período.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro recebido, por meio do programa PICDT, durante parte do período de doutoramento.

A Antonio Lucio Vivaldi (1678* - 1741†), cujas belíssimas composições foram minha companhia quase constante na escrita deste documento.

ABSTRACT

Sensor networks comprise small electro-mechanical devices that communicate over a wireless network. These devices collect environmental data and send them to a remote base station. The main goal of a data collection scheme for sensor networks is to keep the network's database updated while saving the limited nodes' energy as much as possible. To achieve this goal without continuous reporting, data suppression is a key strategy. The basic idea behind data suppression schemes is to send data to the base station only when the nodes' readings are different from what both nodes and base station expect. One alternative of data suppression is to cluster the nodes, aggregate their data and send only a summary to the base station. We propose to group the nodes into spatially homogeneous clusters, which consider both the geographical distance and the similarity of measurements between the neighboring nodes. Through simulated experiments, we have concluded that spatially homogeneous clusters produce data summaries with a higher statistical quality if compared with the usual ordinary clustering methods. Since distributed clustering algorithms play an important role in energy-efficient data collection proposals for sensor networks, we present *Distributed Data-aware Representative Clustering* (DARC) algorithm and *Data-Aware Distributed Clustering Algorithm* (DA-DCA). DARC and DCA build clusters around clusters' representatives, which are able to produce more homogeneous clusters than the usual clustering proposals. Then, they produce data summaries that estimate the nodes' data with a smaller error if compared with the usual data-aware clustering proposals. Another important characteristic of data suppression schemes is their sensitiveness to aberrant readings, since these outlying observations mean a change in the expected behavior for the readings sequence. Transmitting these erroneous readings is a waste of energy. In this thesis, we present a temporal suppression scheme that is robust to aberrant readings. We propose to use a technique to detect outliers from a time series. Since outliers can suggest a distribution change-point or an aberrant reading, our proposal classifies the detected outliers as aberrant readings or change-points using a post-monitoring window. This idea is the basis for a temporal suppression scheme named TS-SOUND (*Temporal Suppression by Statistical OUtlier Notice and Detection*). TS-SOUND detects outliers in the sequence of sensor readings and sends data to the base station only when a change-point is detected. Therefore, TS-SOUND filters aberrant readings and, even when this filter fails, TS-SOUND does not send the deviated reading to the base station. Experiments with real and simulated data have shown that TS-SOUND scheme is more robust to aberrant readings than other temporal suppression schemes proposed in the literature (value-based temporal suppression, PAQ and exponential regression). Furthermore, TS-SOUND has got suppression rates comparable or greater than the rates of the cited schemes, in addition to keeping the prediction errors at acceptable levels.

SUPRESSÃO DE DADOS EM REDES DE SENSORES: MELHORANDO A QUALIDADE DAS ESTIMATIVAS E A ROBUSTEZ A DADOS ABERRANTES

RESUMO

Redes de sensores são formadas por minúsculos componentes eletromecânicos que coletam dados ambientais e os enviam até uma estação-base remota por meio da comunicação sem fio entre os nós-sensores. O principal objetivo de um esquema de coleta de dados para rede de sensores é manter a estação-base atualizada enquanto economiza a maior quantidade de energia possível. Para atingir este objetivo sem um monitoramento contínuo, a supressão de dados é uma estratégia chave. A idéia da supressão de dados é enviar dados para a estação-base somente quando os dados dos nós-sensores forem diferentes do que os nós e a estação-base esperam. Uma forma de suprimir dados é agrupar os nós-sensores, agregar seus dados e enviar somente um resumo para a estação-base. Esta tese propõe agrupar os nós-sensores em conglomerados espacialmente homogêneos, que consideram tanto a distância geográfica e a similaridade de medidas entre nós-sensores vizinhos. Utilizando experimentos simulados, nós concluímos que conglomerados espacialmente homogêneos produzem resumos que possuem uma qualidade estatística melhor se comparados com os resumos produzidos pelos métodos de aglomeração usuais (*ordinary clustering*). Visto que algoritmos de aglomeração distribuídos têm um papel importante na eficiência energética das propostas para coleta de dados em redes de sensores, nós apresentamos o algoritmo *Distributed Data-aware Representative Clustering* (DARC) e o *Data-Aware Distributed Clustering Algorithm* (DA-DCA). Os algoritmos DARC e DCA formam conglomerados de nós-sensores em torno de um representante dos conglomerados, o que gera conglomerados mais homogêneos do que aqueles formados pelas propostas usuais na literatura. Assim, estes conglomerados produzem resumos que estimam os dados dos nós-sensores com um erro menor se comparado aos algoritmos de aglomeração *data-aware* usuais. Outra característica importante dos esquemas de supressão de dados é a sua sensibilidade a dados aberrantes. Estas observações discrepantes significam uma mudança no comportamento esperado para aquela sequência de observações. Transmitir estes dados errôneos para a estação-base é um desperdício de energia. Nesta tese, nós apresentamos um esquema de supressão temporal que é robusto a observações aberrantes. Nós propomos usar uma técnica de detecção de *outliers* em uma série temporal. Visto que *outliers* podem ser um indício de ponto de mudança na série ou uma observação aberrante, nossa proposta é classificar os *outliers* detectados como observação aberrante ou pontos de mudança usando uma janela de pós-monitoramento. Esta é a idéia por trás do TS-SOUND (*Temporal Suppression by Statistical OUtlier Notice and Detection*). TS-SOUND é a nossa proposta para um esquema de supressão temporal de dados. Ele detecta *outliers* na sequência dos dados de um nó-sensor e envia dados para a estação-

base somente quando uma mudança é detectada. Deste modo, o TS-SOUND filtra as observações aberrantes e, mesmo que este filtro falhe, o TS-SOUND não envia a observação discrepante para a estação-base. Experimentos com dados reais e simulados mostraram que o TS-SOUND é mais robusto a observações aberrantes do que outros esquemas de supressão temporal propostos na literatura. Além disto, TS-SOUND consegue taxas de supressão comparáveis ou maiores do que as taxas de supressão de outros esquemas além de manter os erros de predição em níveis aceitáveis.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF ABBREVIATIONS

CV	Coefficient of variation
DARC	Data-Aware Representative Clustering
DA-DCA	Data-Aware Distributed Clustering Algorithm
EXP	Exponential Regression (data suppression scheme)
GPS	Global Positioning System
LEACH	Low-Energy Adaptative Clustering Hierarchy
PAQ	Probabilistic Approximate Querying
MAD	Mean Absolute Deviation
MST	Minimal Spanning Tree
SDAR	Sequentially Discount Auto Regressive
SKATER	Spatial 'K'luster Analysis by Tree Edge Removal
SNAP	Snapshot Queries
TS-SOUND	Temporal Suppression by Statistical Outlier Notice and Detection
VB	Value-based

1 INTRODUCTION

Understanding how the physical world works is a constant concern of the humanity. During centuries, the human beings observe the environment, make questions, conclude and observe again. The study of the environment is an important task to warrant the survival of many species. How to recognize an imminent earthquake without recognizing its signals? But how to know which are such signals without collecting data about earthquakes?

To expand their knowledge about the physical world, environmental researchers have installed observation structures that are able to collect data on several kinds of environments in large geographical areas. In the East of Norway, mechanisms to detect glaciers' movements have been embedded in the ice. Mini meteorological stations have been installed along the stem of high trees to study the microclimate of a forest in Sonoma, CA, EUA. Structures to measure the air quality are present in 23 places in Sao Paulo city, Brazil. There are much many examples, as the programs for earth observation of the National Oceanic Atmospheric Administration[1] and the U.S. Geological Survey[2].

In Brazil, the National Institute for Space Research (INPE) coordinates the *Brazilian System for Environmental Data Collection*. This system comprises the satellites SCD-1, SCD-2 and CBERS-2, a network with 750 platforms for data collection that are spread over the national territory, two reception stations (Cuiabá, MT, and Alcântara, MA) and the *Center for Data Collection Mission* (Centro de Missão Coleta de Dados) in Cachoeira

[1] http://www.noaa.gov
[2] http://www.usgs.gov

Paulista, SP. Figure 1.1A presents the spatial distribution of the *Data Collection Platforms* (PCD, in Portuguese) and Figure 1.1B shows one of these platforms. The satellites SCD-1 and SCD-2 enable the communication among a PCD and the reception stations. A PCD collects meteorological data as air temperature, air relative humidity, atmospheric pressure, wind speed, wind direction, solar radiation, soil temperature and pluviometric precipitation. These structures have large dimensions and are powered by solar batteries.

A) Spatial distribution of the PCD network.

B) Installed PCD.

The colors in A) identify the PCD type: the red dots are the meteorological stations; the blue dots are the hydrometeorological stations and the yellow dots are the agrometeorological stations.
Source: http://tempo.cptec.inpe.br:9080/PCD/

Figure 1.1 - Brazilian System for Environmental Data Collection.

A large number of applications use the data collected by the PCD such as meteorological predictions[3], hydrological models and agriculture studies. Because of its usefulness and dimension, the Brazilian System for Environmental Data Collection is a concrete example of the huge effort of the human society to understand the physical world.

Despite of being a frequent activity, monitoring phenomena in large geographical areas is still a costly and hard task. The observation structures are expensive and require frequent maintenance. Besides, they often have constraints on the deployment, which make unfeasible the proper coverage of the study area. In addition to this, the temporal resolution of the collected data may not be large enough to allow for studying the phenomenon. However, an emerging technology promises to solve these problems and help us to observe the physical world: the sensor networks.

1.1 Sensor Networks

Sensor nodes are minuscule electro-mechanical platforms that comprise sensors, a processor, a radio, memory and batteries (Figure 1.2). The sensors can collect environmental data as air temperature, atmospheric pressure, suspension particles, salinity, air relative humidity, soil moisture, solar radiation and so on. A sensor node collects data, processes them and sends the result to a remote base station using wireless communication among its neighbors. The sensor nodes and the set of wireless links among them form a *sensor network*. If the sensors collect data whose geographical

[3] http://www.cptec.inpe.br

information is important (for instance, localization), they are called *geo*sensors and form a *geosensor network* (NITTEL and STEFANIDIS, 2005).

The wireless communication is an important characteristic of a sensor network. In addition to the small size of its components, the wireless aspect of the sensor netwoks makes them a valuable instrument for collecting data without being invasive or disturbing. This is especially useful to study ecosystems and wild life, which are sensitive to the human presence.

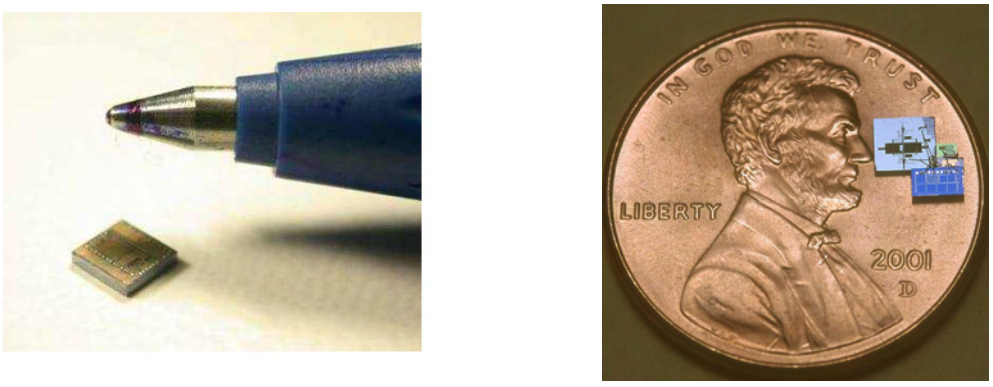


Figure 1.2 - Sensor nodes. Left side: "Spec" mote (by University of California- Berkeley). Rigth side: Golem Dust, one of the generations of sensors of the Smart Dust Project.

The research in sensor networks has been motivated by military applications, as the surveillance systems for the oceans and the networks to detect ground targets. However, the recent technological advances have decreased the production costs and increased the capacity of the mechanisms. This has contributed to widen the range of applications for sensor networks. Among them, we have the detection of natural disasters (earthquakes, seaquakes, volcanic eruptions) and non-natural disasters (biological contamination, oil spilling, fires), habitat monitoring, traffic organization and smart environments.

In spite of the rapid advances in the technological development of the sensors mechanisms, the sensor networks installed until now have been part of experiments. This is the case of the projects *Great Duck Island* and ZebraNet.

The *Great Duck Island* Project (Mainwaring et. al, 2002) began in August, 2002, when researchers of the University of California (Berkeley) and the *Intel Research Laboratories* installed a sensor network in *Great Duck Island*, Maine (EUA). The goal was to monitoring the behavior and the habitat of a bird named *Storm Petrel*. Initially, 32 sensor nodes (Figure 1.3) were deployed close to the birds' burrows. Posteriorly, the researcher enlarged the network adding more nodes and meteorological stations. The sensor nodes collected data on air temperature and relative humidity, atmospheric pressure and other variables. The periodic nodes readings were transmitted to a special node named *gateway* through the wireless communication among the nodes. The gateway re-transmitted the readings to the base station, which stored the readings. At each 15 minutes, a copy of the database was transmitted, through a satellite, to the server at the University in Berkeley. The users had access to database through the internet[4].

In ZebraNet Project[5] (JUANG *et al.*, 2002), sensor necklaces with GPS (*Global Positioning System*) were installed in zebras (Figure 1.3) of the natural reserve of *Sweetwaters*, Republic of Kenia. From the biological point of view, the goal of the project was to monitor the nocturnal behavior of the animals and answer some questions about migration and

[4] http://www.greatduckisland.net/
[5] http://www.princeton.edu/%7Emrm/zebranet.html

relationships inter-species. The sensors were designed to register and store the animals' position every 3 minutes. At each one hour, meteorological and environmental data, luminosity and temperature data, in addition to body movements, were registered during 3 minutes. As the nodes platforms (the zebras), the base station (a laptop computer) was also mobile and often went through the sensors field (zebras' *habitat*) to gather the sensed data.

Figure 1.3 - Sensor nodes deployed by the projects *Great Duck Island* and *ZebraNet* Project (left and right side, respectively). Left side: MICA-2 (by Intel and University of California- Berkeley). Rigth side: Sensor necklace.

Although promising to be a powerfull instrument for pervasive and non-disturbing data collection, sensor networks are a constrained environment. They have limitations in the data processing, communication range, data storage and, mainly, in the energy. Sensor nodes carry a limited amount of energy, which is used to do all their tasks. Once a sensor network is deployed, its goal is to operate with a minimum or no human attendance.

The data transmission faces the main limitation of a sensor network: the energy consumption. The communication task spends much energy, more than processing and collecting data (POTTIE and KAISER, 2000). As a result, the data collection protocols for sensor networks are an increasing research field and a large number of protocols has been proposed (AKKAYA and

YOUNIS, 2004). These proposals present several strategies to reduce the energy spending.

The data delivery model of a sensor network depends on its application. Tilak et al. (2002) have identified three data delivery models: continuous data collection and delivery; continuous data collection but the data delivery is triggered by pre-defined events; and on-demand data collection and delivery (queries). The first data delivery model is suitable to applications that require continuous updates of the database at the network's base station, such as environmental and habitat monitoring. In these applications, researches are usually interested in observing phenomena that require data with high temporal resolution to be completely understood, such as changes in the micro-climate of a forest (TOLLE *et al.*, 2005), seismic waves along an active volcano (WERNER-ALLEN *et al.*, 2006), soil moisture recharge along the roots of a tree (RYEL *et al.*, 2003), glaciers movement (PADHY *et al.*, 2005) and so on. In this thesis, we are interested in these types of applications.

Since sending continuous reports would quickly run out the limited energy of the nodes, sensor networks designed for environmental and habitat monitoring have few alternatives to save energy in their data routing. One of these alternatives is to use *data suppression*.

1.2 Data suppression to collect data in a sensor network

To define a data suppression scheme, nodes and base station have to agree on an expected behavior for the nodes' readings. Therefore, nodes only send reports to the base station when their sensed values do not agree with the expected behavior. Otherwise, nodes suppress their data. If the base station does not receive any data from a node, it uses the expected behavior to predict the suppressed data.

In a temporal suppression scheme, a node decides when to suppress or not its data. In a given time period, one of three events can occur: a) all nodes decide to suppress their data; b) all nodes decide to send their data to the base station or c) some nodes send data while the remaining nodes decide to suppress their data. A temporal suppression scheme uses the correlation among the readings of a same node to build the expected behavior for the nodes' readings (TULONE and MADDEN, 2006). In a spatial suppression scheme, at each time period, some nodes are allowed to send their data while the remaining nodes must suppres their data (KOTIDIS, 2005). Differently from the temporal scheme, the base station receives data in all time periods, but not from all nodes. A spatial suppression scheme considers the correlation among the observations of neighboring nodes (SILBERSTEIN *et al.*, 2007a). Finally, a spatio-temporal suppression scheme considers both types of correlations, temporal and spatial, to build the expected behavior of the sensed data (SILBERSTEIN *et al.*, 2007a).

Another strategy to save energy in sensor networks that have to keep the base station continuously updated is to *cluster the nodes and aggregate their data*. Before each transmission, nodes form clusters around a node chosen as their cluster head. The head node receives data from its cluster members, aggregates these data and sends only a summary to the base station. From now on, we call this strategy *cluster-and-aggregate*. It localizes the communication among nodes, reduces the messages volume traveling through the network and, as a result, reduces the energy spending. Moreover, data processing spends less energy than the communication among the nodes (POTTIE and KAISER, 2000).

We can consider the cluster-and-aggregate strategy as a spatial data suppression scheme, since the nodes of a cluster, except by the head, do not transmit their data. The base station agrees on estimating the suppressed data using the summary received from the cluster head.

Both data suppression and data aggregation (in cluster or along the routing path) can be considered to be instances of *Information (or Data) Fusion.* In general, both terms (data and information) are accepted. Widening the discussion about a definition for Data Fusion, Wald (1999) defines data fusion as "a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend upon the application". In the sensor network context, one can use information fusion with at least two objectives: to improve data accuracy and/or save energy (NAKAMURA *et al.*, 2007).

In the information fusion context, data aggregation represents the instance of summarization. It allows for energy saving, although it means accuracy loss (NAKAMURA *et al.*, 2007). Since "any processing of time-series of data acquired by the same sensor or different sensors is a fusion process" (WALD, 1999), a temporal data suppression can be also considered as an instance of information fusion.

1.3 Problem Definition

Several schemes have been proposed to achieve energy saving by data suppression (such as Kotidis (2005), Chu *et al.* (2006), Tulone and Madden (2006), Silberstein et al. (2007a)) and data aggregation (such as Subramanian and Katz (2000), Manjeshwar and Agrawal (2001), Heinzelman et al. (2002), Lindsey and Raghavendra (2002), Younis et al. (2002), Singh and Gore (2005) and those described by Akkaya and Younis (2004)).

The energy saving is the most common metric to evaluate the efficiency of a data collection protocol. However, in the case of sensor networks that must continuously update the base station, both strategies to save energy

(data suppression and data aggregation) lead to estimates for the sensed data. Therefore, we argue that *statistical quality of these estimates should be also used to evaluate the performance of a data collection scheme*. In this thesis, we evaluate the statistical quality of an estimate using the estimation error, which is defined as the difference between the real value and its estimate. Besides, we propose a distributed clustering algorithm to produce summaries that are better estimates for the aggregated data if compared with the usual clustering proposals. This clustering algorithm can be used the basis for a spatial suppression or spatio-temporal suppression scheme.

Although being a key strategy to get continuous updating without continuous reporting (SILBERSTEIN *et al.*, 2007a), data suppression schemes are sensitive to aberrant readings. The suppression scheme interprets these erroneous values as a change in the expected behavior and nodes send an outlying reading to the base station. This means a waste of energy and, possibly, a bad updating of the expected behavior.

Sensors measuring environmental variables produce nonsense readings as a result of temporarily malfunctioning or due to some intervention on the monitored environment that is not related to the monitored variables. In regular weather stations, which have low energy constrains, nodes transmit or record the aberrant readings, which are identified and deleted at the base station. However, for the constrained environment of a sensor network, transmitting nonsense values means to waste valuable resources. As a solution to this problem, this thesis proposes a temporal data suppression scheme to be robust to aberrant readings.

1.4 Hypotheses, objectives and main contributions

In this thesis, we see the sensor network as an instrument for sampling spatio-temporal phenomena, collecting spatio-temporal data. By data collection, we mean all the three steps: sensing, processing and transmitting. The product of a data collection is the database at the base station.

For an important set of applications, the constraints of a sensor network does not allow for getting the real sensed data at the base station. Frequently, the network's database is updated by estimates of the real sensed data. Trying to improve the quality of these estimates, the following hypotheses are considered:

1) The statistical quality of the summaries produced by the cluster-and-aggregate strategy can be improved if the clusters are spatially homogeneous;

2) The energy saving, as well as the statistical quality of the updates in the network's base station, can be improved if we use a temporal data suppression scheme that is robust to aberrant readings.

The main goals of this thesis are twofold:

1) Propose and evaluate a distributed clustering algorithm to produce summaries that are better estimates for the individual data if compared with the usual clustering proposals.

2) Propose and evaluate a temporal data suppression scheme to be robust to aberrant readings.

Besides the proposals described above, the main contribution of this thesis is to introduce the statistical quality of the estimates delivered to the

network's user as an additional metric to evaluate the performance of a data collection proposal for a sensor network.

1.5 Thesis Layout

This thesis comprises three papers on the topics discussed above and is structured as follows:

a) Chapter 2 examines the effect of using a data-aware clustering procedure on the statistical quality of the data received by the base station. This chapter examines the first hypothesis.

b) Based on the findings of Chapter 2, Chapter 3 presents a proposal for distributed clustering algorithm to obtain more homogeneous clusters of nodes.

c) Chapter 4 presents the proposal for a temporal data suppression scheme that is robust to aberrant readings. This chapter examines the second hypothesis.

d) Chapter 5 presents the concluding remarks and points to future works.

2 DATA-AWARE CLUSTERING FOR GEOSENSOR DATA COLLECTION♦

The advances in wireless and miniaturisation technologies are making possible the development of the sensor networks, a new instrument for the remote sensing of the physical world (ELSON and ESTRIN, 2004).

Sensor networks are composed by a large number of small nodes. These nodes are electro-mechanical devices that measure environmental characteristics such as temperature, pressure, humidity and luminosity. These data are disseminated through wireless communication among the nodes until a base station is reached. Once sensor networks are deployed in the study region, they work without human attendance.

The environmental monitoring is one of many potential uses of this emerging technology (XU, 2002), especially for hostile environments. According to Martinez et al. (2004), the sensor networks will make possible a realistic monitoring of the natural environmental. In their work, the authors discuss how the environmental monitoring evolved from data logging to sensor networks and describe the GlacsWeb project, an ongoing research in subglacial bed deformation.

Other environmental applications involving sensor networks are described in the literature. Among them, we have the monitoring of the environment of rare and endangered species of plants in a volcano neighboring (BIAGIONI and BRIDGES, 2002); the monitoring of the habitat of seabirds (MAINWARING *et al.*, 2002); the microclimate monitoring throughout the volume of giant trees (CULLER *et al.*, 2004); the flood monitoring to provide warnings and the monitoring of coastal erosion around small

♦ This chapter is an adaptation of the work in REIS *et al.* (2007).

islands (ENVISENSE-SECOAS). Until recently, experiments have been run on small-scale sensor networks and no large-scale networks have yet been deployed in practice. However, as the sensors become smaller and cheaper (WARNEKE *et al.*, 2001), sensor nodes are expected to be densely deployed in the environment.

Some sensor networks are designed to collect data whose geospatial information is important. To stress their geographic characteristic, these networks are usually defined as geosensor networks (NITTEL and STEFANIDIS, 2005). The main goal of a geosensor network is to collect geospatial data while keeping the energy consumption at an acceptable level.

Geosensor networks are an application-driven technology. The temporal resolution of the data determines their delivery model while the required spatial resolution defines the degree of data summarization. Tilak et al. (2002) have identified three data delivery models: continuous data collection and delivery; continuous data collection but the data delivery is triggered by pre-defined events; and on-demand data collection and delivery (queries). On the spatial resolution, some applications need the raw data of all sensing points (CHU *et al.*, 2006,TULONE and MADDEN, 2006), whereas others need just a summary of all sensors' data, as those TAG (Tiny AGgregation) (MADDEN *et al.*, 2002) has been designed for. In the middle of these two extreme cases, there are applications that accept an intermediate degree of data summarization, as maps of temperature and relative humidity, for instance. These applications have goals as identifying zones of interest such as hot and cold zones. Sensors' data are summarized over subregions, pre-defined (GOLDIN, 2006) or not, and the spatial distribution of these summaries provides a report of the data variability over the entire region.

In this chapter, we are interested in applications that require a continuous data delivery and admit an intermediate degree of data summarization.

For continuous data delivery, hierarchical cluster-based data routing protocols are considered to be the most energy efficient alternative (HEINZELMAN *et al.*, 2002). Multiple cluster-based protocols as LEACH and LEACH-C (HEINZELMAN *et al.*, 2002) are suitable for applications that admit data summaries over subregions of the sensor field. A cluster-based protocol assembles the sensor nodes into clusters before the data transmission. Except for clustering procedures as those in Kotidis (2005) and Tulone and Madden (2006), the usual clustering algorithms consider only the nodes closeness, which we define as *ordinary spatial clustering*. A node chosen as the cluster head receives data from all nodes in its cluster, aggregates these data and sends the summary to the base station. Clustering the nodes keeps most of the communication inside the clusters while data aggregation reduces the messages volume travelling through the network. These strategies together allow for energy saving.

Data aggregation presumes nearby nodes have correlated data. Thus, they are similar to each other and one can aggregate the nodes' data of an ordinary spatial cluster to represent this cluster.

We agree with this reasoning but we believe presuming data correlation is not enough to produce data summaries that are the best estimates of the summarized data. A partition of nodes that considers only their geographical location is missing the most important: the measurements themselves. To make our point clear, consider Figure 2.1, which presents the spatial distribution of luminosity measurements, for instance. Suppose we regularly deploy a geosensor network in the region. The area delimited at right bottom corner has a lower spatial variability in its measurements than the delimited area at upper left corner. Suppose we use one single cluster to summarize the data of each area. A data summary as the

average, for example, estimates better the summarized data in the area at right bottom corner than in the area at upper left corner. Besides, a single cluster could summarize the data in the first area whereas the second area would require a larger number of clusters, to account for the increased spatial variability. To capture these different requirements, the nodes partition might consider the nodes measurements in addition to their geographical location.

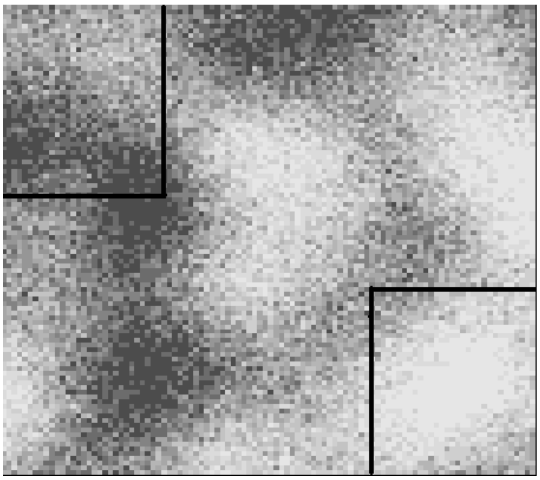


Figure 2.1 - Spatial distribution of luminosity measurements.
The delimited areas present different spatial variability in their measurements.

Based on these considerations, we present the contributions of this chapter.

2.1 Our proposal

We propose a data-aware clustering procedure that groups the nodes considering the spatial homogeneity of the nodes' data in addition to their location. Our hypothesis is that data summaries based on spatially homogeneous clusters will have a better statistical quality if compared with data summaries based on ordinary spatial clusters. A statistical quality

measure expresses how well the data summary sent to the base station estimates the data collected by the nodes.

In this chapter, our major concern is to examine how the spatial arrangement of the clusters in a geosensor network affects the statistical quality of the data received by the base station. We compare spatially homogeneous clusters with ordinary spatial clusters regarding the statistical quality of their summaries.

The remainder of this chapter is organized as follows. Section 2.2 describes some proposals for data routing based on clusters and selects those suitable to our purposes. In section 2.3, we define the spatially homogeneous clusters and give a brief description of SKATER, the procedure for obtaining such clusters. In Section 2.4, we present the data processing steps and the types of data produced when one samples a spatial dataset using geosensor networks with a multiple cluster-based routing protocol. Section 2.5 presents the main results of simulated experiments comparing ordinary and spatially homogeneous clusters, based on the statistical quality of their data summaries. Finally, section 2.6 draws some concluding remarks.

2.2 Cluster-based proposals to route sensor data

In this section, we describe the clustering procedures of the main cluster-based data routing protocols and select those suitable to our purposes. Considering the applications we are interested in this chapter, the suitable protocols must have a continuous data delivery and produce multiple cluster summaries.

Ibriq and Mahgoub (2004) have observed that the usual cluster-based data routing protocols just consider the spatial closeness of the nodes and their

energy reserve to create the clusters. We define these clusters as ordinary spatial clusters.

One of the first cluster-based data routing protocols developed for sensor networks was LEACH (Low-Energy Adaptative Clustering Hierarchy) (HEINZELMAN *et al.*, 2000). LEACH´s clustering procedure has a distributed algorithm that uses an estimate of the energy level of the nodes to choose the cluster head nodes. To assemble the clusters, LEACH uses the strength of communication between the cluster head and the other nodes of the cluster as its closeness measure. Since LEACH´s clustering procedure has no control over the cluster heads distribution in the study region, Heinzelman et al. (2002) proposed the LEACH-C protocol. The clustering procedure of LEACH-C uses a centralized algorithm at the base station that tries to optimize the clusters distribution over the study area. This optimization algorithm needs the energy level of the nodes as well as their location. As a result, LEACH-C is suitable for geosensors networks and its clusters are better distributed over the network area than LEACH clusters.

We classify the clustering procedures of LEACH and LEACH-C as ordinary spatial clustering. Both protocols have a continuous data delivery model and produce multiple cluster summaries. Therefore, LEACH and LEACH-C are suitable to the purposes of this chapter.

Many cluster-based data routing protocols have been proposed as versions of LEACH, such as PEGASIS (LINDSEY and RAGHAVENDRA, 2002), TEEN (MANJESHWAR and AGRAWAL, 2001) and their improvements. In contrast to multiple cluster-based procedures of LEACH and LEACH-C, these protocols have clustering procedures that produce a summary of all sensors´ data. Since we are interested in applications that admit an intermediate degree of data summarization, PEGASIS and TEEN are not suitable to the applications we are interested in this chapter.

Some cluster-based data routing protocols were developed independently of LEACH, such as the proposals of Younis et al. (2002) and Subramanian and Katz (2000). In contrast to LEACH, these protocols were designed to networks that have supernodes. These special nodes are richer in energy, computational and communication resources than the other ones. The supernodes are the cluster heads, being responsible for data aggregation and data routing. Despite being a smart solution for the problem of energy constraint, a network with heterogeneous nodes creates a new constraint. The nodes' deployment has to be controlled to avoid agglomerations of supernodes. This constraint reduces the ease of deployment, one of the wanted properties of a sensor network (HEINZELMAN *et al.*, 2002). Therefore, we have not considered these protocols in this chapter.

An extensive survey of cluster-based data routing protocols for sensor networks is not our aim. For a review of this subject, we refer to the works of Ibriq and Mahgoub (2004) and Akkaya and Younis (2004).

2.3 Spatially Homogeneous Clusters

In contrast to ordinary spatial clusters, the definition of spatially homogeneous clusters considers explicitly the nodes' attributes besides their geographical location (ASSUNÇÃO *et al.*, 2006). Spatially homogeneous clusters are clusters resulting from a partition procedure with three properties.

First, nodes belonging to same cluster have to be similar to each other in some predefined attributes (cluster internal homogeneity). Second, nodes belonging to different clusters have to be different from each other

(heterogeneity among clusters). Third, the nodes of a same cluster must belong to a predefined neighborhood structure (closeness or contiguity). The clustering proposals in Kotidis (2005) and Tulone and Madden (2006) assemble clusters around representatives nodes[6] based on the similarity between a representative node and the nodes inside of its range of communication. However, there is no warranty the first and second properties are satisfied. So, they cannot be classified as spatially homogeneous clusters.

To get the spatially homogeneous clusters, we propose the use of the spatial clustering algorithm SKATER (*Spatial 'K'luster Analysis by Tree Edge Removal*) (ASSUNÇÃO *et al.*, 2006). This algorithm is a strategy for transforming the regionalisation problem into a graph partitioning problem. SKATER works in two steps. First, it creates a minimal spanning tree (MST) from the graph representation for the neighborhood structure of the geographic entities. The cost of an edge represents the similarity of the entities' attributes, defined as the Euclidean squared distance between them. The MST represents a statistical summary of the neighborhood graph based on the entities' attributes. In the second step, SKATER performs a recursive partitioning of the MST to get contiguous clusters. The MST partitioning considers explicitly the clusters internal homogeneity.

In the geosensor networks context, the graph vertices are the sensor nodes, the edges are the radio links and the cost of an edge connecting a pair of vertices is the similarity between the nodes' data.

[6] We discuss these two clustering proposals in Chapter 3.

Spatially homogeneous clustering offers the possibility of transforming the undelivered raw data into information, since its clusters represent the partition of the sensor field that has great internal homogeneity regarding the values of monitored variable. This information cannot be directly extracted from the summaries based on ordinary spatial clusters or on clusters as those proposed in Tulone and Madden (2006), for example. As a result, spatially homogeneous clustering can be seen as a tool for spatial sensor data mining.

2.4 Data Processing Steps of a Multiple Cluster-Based Routing Protocol

We see geosensor networks as instruments to sample spatio-temporal data. The protocol that delivers these samples to the final user is a part of this sampling instrument. Geosensor networks that use a multiple cluster-based routing protocol submit the collected raw data to two processing steps: data sampling and data aggregation. The main goal of this section is to define these data processing steps as well as their input and output data. Defining these processing steps is important to figure out the task of a geosensor network that employs a cluster-based protocol to route its data. Here, we suppose the final product of the geosensor network, which is the data delivered at base station, will be used to identify zones with extreme values.

Figure 2.2 illustrates the path the data follow from the sensors field to the base station. To define the rawest type of data, we suppose the study region is divided into subregions with the same size, which is the smallest possible or suitable. The measurements taken in these subregions are called the *original data*. These data are the observation of a spatial variable. We can see the spatial distribution of the original data as an image of the phenomenon that has the best resolution possible.

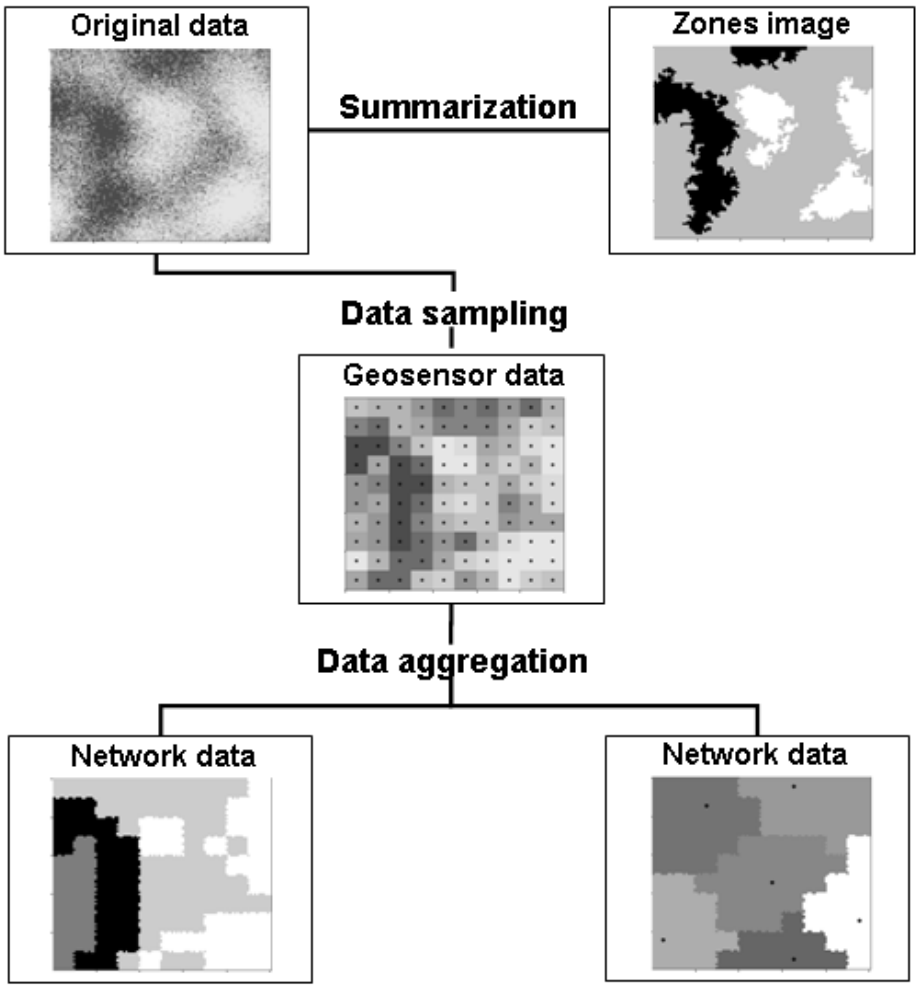


Figure 2.2 - Data processing during their path from the sensors field to the base station. The network data were obtained by the spatially homogeneous clustering (left box) and ordinary clustering (right box).

Collecting the original data is impracticable, since this means to take measurements in the whole study region. However, the analysts are not usually interested in the raw data. Even if they had the original data, some summarization will be necessary to understand the phenomenon these data represent. The analysis goal determines the type of summarization that must be done. Suppose our interest is to detect zones with extreme values. If we measure temperature, for example, we want to detect cold and hot zones. For luminosity measures, we are interested in dark and clear zones.

We define a *zones image* as the summarization of the original data that delimit the zones with extreme values (Figure 2.2).

Sampling the original data is the first data processing. As the geosensor network is the sampling instrument, we call the resulting spatial sample as *geosensor data*. If the network is regularly deployed, we can see the spatial distribution of the geosensor data as a phenomenon image with a smaller resolution (Figure 2.2).

Data aggregation is the second data processing. The geosensor data are aggregated in clusters and the resulting summary is sent to the base station. The spatial clusters and their summaries are what we define as *network data*. Figure 2.2 shows the network data produced by two alternatives of clustering: spatially homogeneous clustering and ordinary clustering. The two network data are two possible summaries of the geosensor data. To build the visualization of the network data as in Figure 2.2, the base station has also to know the nodes position and which nodes compose each cluster.

Considering the analysis goal is to identify zones with extreme values, we would like to have an adequate summary of the zones image. When we use a geosensor network to sample the original data and a cluster-based protocol to route this sample, we try to produce a summary of the zones image based on a subset of the original data (the geosensor data). In other words, the cluster-based protocol tries to reproduce a summarization of the phenomenon image just looking at a version of this image with a smaller resolution. Having these considerations in mind, the analysis of the quality of the network data must consider how difficult is to reproduce the zones image of a spatial dataset based on a sample of this dataset.

2.5 Assessing the Performance of the Spatially Homogeneous Clusters

We have carried out simulation experiments to evaluate and compare the statistical quality of data summaries based on spatially homogeneous clusters and ordinary spatial clusters. Some results were not presented here for briefness.

2.5.1 The simulated experiments

We have simulated datasets with spatial autocorrelation using a grid of 10000 cells (100 x 100). For practical reasons, we have considered each cell as a square of side 1 meter. We refer these data as *original data*. These datasets were characterized by *extreme zones*, which are groups of cells with high values (clear zones) and groups of cells with low values (dark zones).

The zones size was controlled by a scale parameter. Figure 2.1 presents the spatial distribution of a dataset simulated using a scale parameter equal to 20. The left column of the Figure 2.3 presents some examples of original data that we have simulated using the values 5, 10 and 15 for the scale parameters. The higher the scale value is, the larger the zone size. For each value of the scale parameter, we have simulated 500 spatial datasets.

The model adopted for the covariance function was the Gaussian model, in which the covariance value between two locations decreases with the squared distance between them. The data probabilistic model was the Gaussian with mean 100 and variance 2. These values were kept constant during all experiments. We have carried out the simulations of the spatial dataset using the package RandomFields (SCHLATHER, 2001) in the R environment (R DEVELOPMENT CORE TEAM, 2005).

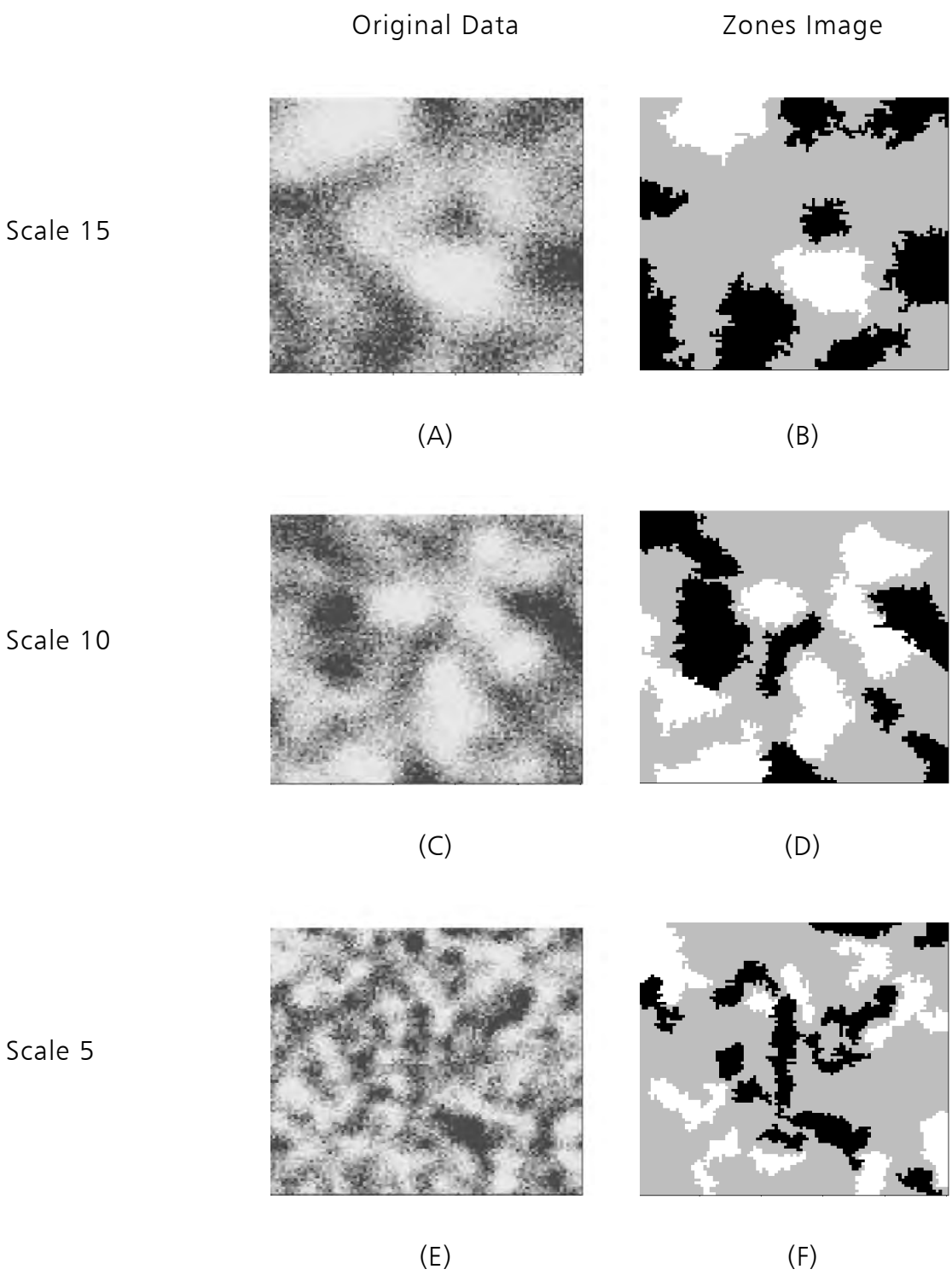


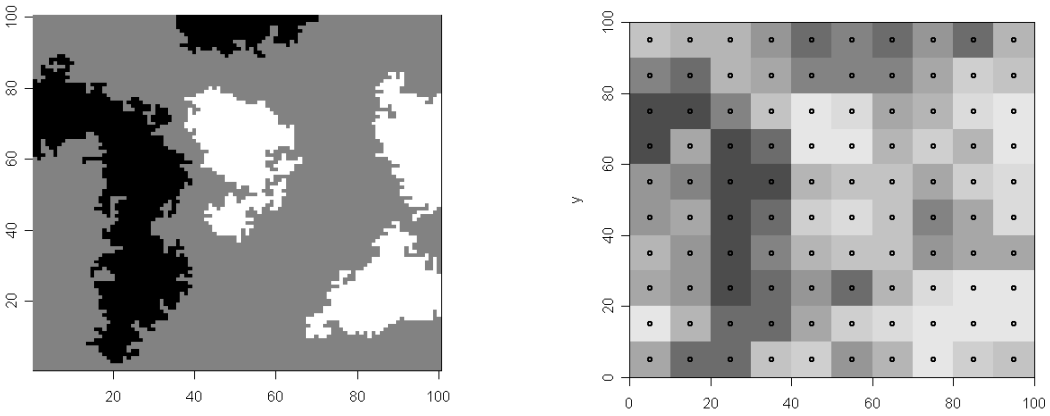
Figure 2.3 - Examples of original data and zones image for three scales.

Dark zones are clusters of cells with values between the 10% smallest and clear zones are clusters of cells with values between the 10% largest. To delimit these zones, we have used image classification techniques, generating the *zones image* (Figure 2.2, right upper corner). We have treated each spatial dataset as an image. Using the geographical information system SPRING (CAMARA *et al.*, 1996), each image was segmented by a region growing algorithm (BINS *et al.*, 1996). Posteriorly, we have classified the image segments into dark and clear zones using the supervised classification technique based on the Bhattacharya distance. We have accepted a classification result if it satisfied two conditions. First, the average value of the image cells classified as clear had to belong to the 10% largest image values. Second, the average value of the image cells classified as dark had to belong to the 10% smallest image values.

The result was the *zones image*. The right column of the Figure 2.3 presents the zone images for the original data in the left column of the same figure. We have used these images to evaluate the ability of the clustering methods to identify the extreme zones.

To sample the spatial datasets, we have deployed a geosensor network with 100 nodes in a regular fashion, as illustrated by the black dots at the right side of the Figure 2.4. Each sensor node has an *area of influence*, which is the area around the node. In our experiments, we have defined the *area of influence of a sensor node* as the set of cells of which the node was the nearest node. The data collected by a sensor node were defined as the

average of the values of the cells in its area of influence[7]. We refer to this sample as *geosensor data*.



Zones image Geosensor data and network's nodes

Figure 2.4 - Zones image and geosensor data of the dataset in Figure 2.1.

To choose the number of clusters to be used in the nodes partitions, we have adopted the expression proposed by Heinzelman *et al.* (2002). This expression finds the optimal number of clusters that minimizes the total energy dissipated in a data transmission of the LEACH protocol. It involves radio energy model parameters and network parameters (area dimensions, number of nodes and cluster heads distance to the base station). The latter parameter depends on the base station position. Then, we have chosen this position so that base station and at least 75% of all nodes was less than 100 meter apart. This is the radio range of commercial sensors platforms like Crossbow MICA2 mote[8] and Mote^{IV} Telos[9], for example. In our experiments, the base station was placed on position (x=120; y=50) and

[7] In REIS *et al*. (2007), we have adopted another definition for the geosensor data. The data collected by a sensor node were defined as the value of the cell in which the node was placed.
[8] http://www.xbow.com/
[9] http://www.moteiv.com/

its average distance to the nodes is 78.6 meter. Finally, adopting the radio energy model as in Heinzelman *et al.* (2002), the optimal number of clusters is six.

To get the ordinary spatial clusters, we have simulated LEACH's clustering procedure (HEINZELMAN *et al.*, 2002). We have chosen the cluster heads randomly among all nodes, but constrained to a minimal distance of 30 meter between them. This constraint tries to simulate the choice of cluster heads by LEACH-C (HEINZELMAN *et al.*, 2002), avoiding to agglomerate the head nodes. To assemble the clusters, we have associated the remaining nodes to their nearest cluster head.

The spatially homogeneous clusters were obtained by the SKATER procedure (ASSUNÇÃO *et al.*, 2006).

To each cluster k, we have calculated the cluster summary $\mathbf{CM}_k$ as the average of the data of the nodes belonging to the cluster k. The set of $\mathbf{CM}_k$ values of a dataset is the *network data* (Figure 2.2).

We have defined the statistical quality of a cluster summary (SQ_k) using the following expression

$$SQ_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \frac{\left| v_{ik} - \mathbf{CM}_k \right|}{v_{ik}}, \qquad (2.1)$$

where v_{ik} is the original data of the cell i that belongs to the union of the area of influence of all nodes of the cluster k ; N_k is the number of cells belonging to the area of influence of the cluster k. Since we have simulated values larger than zero, $v_{ik} \neq 0$, for all i and k. The statistical quality measure SQ_k is the average of the absolute value of the relative errors of the cluster mean $\mathbf{CM}_k$. It measures how far the cluster summary $\mathbf{CM}_k$ is from the original data, in average, when these original data are replaced by $\mathbf{CM}_k$. The smaller the values of SQ_k, the better the cluster summary $\mathbf{CM}_k$

represents the individual cell values. SQ_k is a measure of the local performance of the clusters summaries.

Differently from the arrangement of clusters obtained by SKATER, there more than one possible arrangement if we use LEACH's clustering algorithm. The clusters' arrangement depends on the location of the chosen heads and, as result, this can affect the performance of LEACH's clusters. Then, to minimize this problem, we have used 10 different cluster heads arrangements for each spatial dataset. To choose one of these arrangements, we have sorted them using their value for $SQ_k^{(max)}$, which is the maximum value for SQ_k,. Then, we have chosen the arrangement that had the smallest value for $SQ_k^{(max)}$. It is worth to note that this procedure favours LEACH's clusters.

To enable the comparison of the performance of both clustering proposals' considering a given spatial dataset, we have summarized the SQ_k values, k = 1,2,...,6, for each dataset t, t = 1,2,3,..., 500, and calcuted the following ratio

$$SQ_t^{Ratio} = \frac{median_{(k=1,2,..,6)}\left(SQ_{kt}^{(SH)}\right)}{median_{(k=1,2,..,6)}\left(SQ_{kt}^{(OS)}\right)},\ t = 1,2,3,...,499,500, \qquad (2.2)$$

where $SQ_{kt}^{(SH)}$ and $SQ_{kt}^{(OS)}$ are the values of the expression in (2.1) for the *k-th* cluster assembled by SKATER and LEACH, respectively, using the *t-th* dataset. If $SQ_t^{Ratio} < 1$, for example, it means SKATER has performed better than LEACH, in median, considering the original data came from the dataset t.

2.5.2 The results

We have evaluated the SQ_k values of the spatially homogeneous (SH) and the ordinary spatial (OS) partitions. Figure 2.5 presents the boxplots[10] for SQ_k values according to the scale parameter. The notch in the lateral borders of the boxplots works as a 95% confidence interval for the median (MCGILL *et al.*, 1978).

Spatial homogeneous clusters have overperformed LEACH's ordinary clusters in all evaluated scales. SH clusters have had the smallest SQ_k values, in median, in addition to the smallest variability. As expected, the larger the scale parameter, the higher is the statistical quality of the clusters.

The clusters produced by both clustering methods have a good statistical quality. The cluster means have had relative errors smaller than 10%, in average (SQ_k values smaller than 0.10)[11]. Analyzing these values, it is worth to remember the hard task of the network data when estimating the original data, as we have described in section 2.4. Moreover, as well noted by FRERY *et al.* (2008), SQ_k "is a 'pessimistic' measure", since the original data is never available in practice. However, "it provides an idea of the error introduced by the whole observation process".

[10] The bottom and the top of the box represent the percentiles 25 and 75, respectively. Therefore, the box's height is a measure of the data variability. The line drawn across the box represents the median and the points outside the dashed lines represent the outlier values. The maximum length of the dashed lines depends on the box's height. If there are not outliers, the ends of the inferior and superior dashed lines represent the minimum and the maximum values, respectively (TUKEY, 1977).

[11] The values of SQ_k have been much smaller than the values obtained in REIS *et. al* (2007). This means that the way we use to define the geosensor data can affect the statistical quality of clustering procedures, as suggested by Prof. Alejandro Frery.

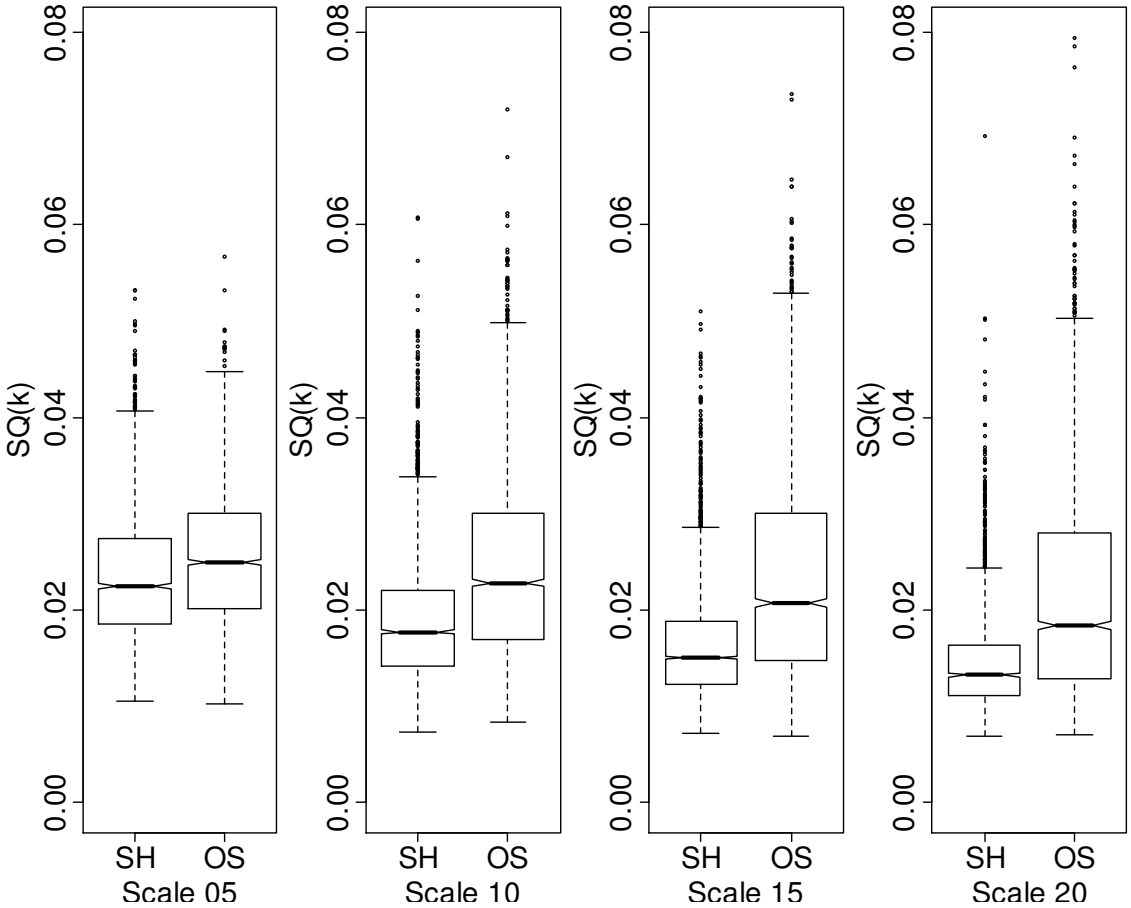


Figure 2.5 - Boxplots for SQ_k values (based on 500 x 6 values).

The boxplots in Figure 2.6 summarize the results for comparison between SKATER and LEACH using the ratio of their median SQ_k values for each spatial dataset. Considering all simulated datasets, SKATER has got values for the statistical quality measure 10% to 30% better, in median, if compared to LEACH. Except by some outliers (less than 10 whatever the scale), SKATER has outperformed LEACH in most of the datasets.

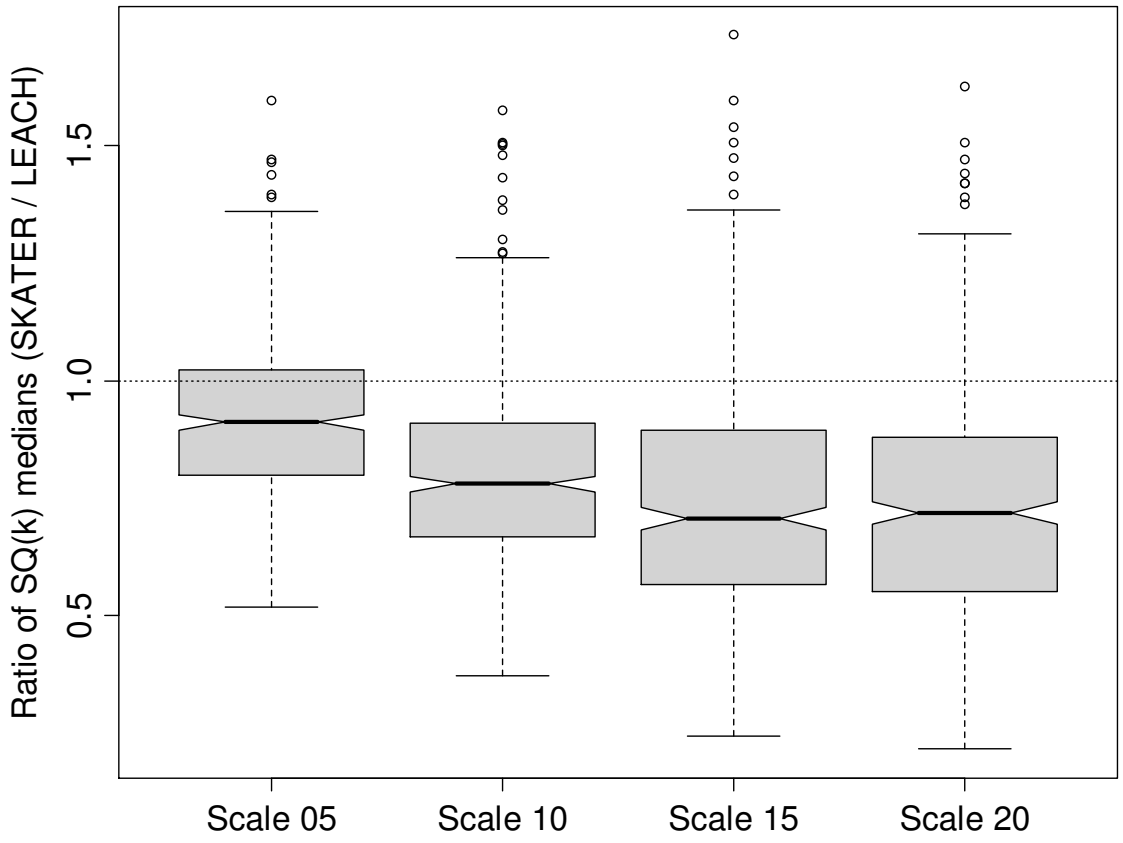


Figure 2.6 - Boxplots for SQ_t^{Ratio} values (based on 500 values).

We have used the summaries $\mathbf{CM}_k$ to classify the clusters into clear or dark, using a criterion similar to that used to produce the zones image. According to its $\mathbf{CM}_k$ value, the cluster *k* was classified as clear ($\mathbf{CM}_k$ value between the 15% largest values of the original data), *dark* ($\mathbf{CM}_k$ value between the 15% smallest values) or *intermediate* (other $\mathbf{CM}_k$ values). We have adopted the cut point 15%, instead of the 10% used for the zones classification, because this value allowed for flexibility in the cluster classification procedure. A cluster having only some nodes out of the "10% zone" could still be classified as an extreme cluster (clear or dark).

Figure 2.7 presents the results of this classification for the two clustering methods applied to the geosensor data sampled from the original data in Figure 2.3 (scale parameter equal to 5, 10 and 15).

Figure 2.8 presents the network data according to two clustering methods applied to the geosensor data of the Figure 2.4 (scale parameter equal to 20). The black clusters are those classified as dark, the white clusters represent the clear clusters and the gray clusters are the intermediate ones. Comparing the zones image in figures 2.3 and 2.4 with the visualization of the network data in figure 2.7 and 2.8, we see the spatially homogeneous clusters could identify more extreme spatial zones than the ordinary spatial clusters. Spatially homogeneous clusters were able to identify zones even when they were small and the geosensor data did not seem to reveal many aspects of spatial autocorrelation (scale parameter equal to 5).

2.6 Concluding Remarks

Within a few years, miniaturized and networked sensors will have the potential to be embedded in several kinds of environments and allow a continuous monitoring (ELSON and ESTRIN, 2004). Geosensor networks will produce a revolution in our understanding of the environment by providing observations at temporal and spatial scales that are not currently possible. Deciding how these data will be routed to the base station is crucial, since data routing is an important consumer of energy, the most critical resource of the network.

The main contributions of this chapter are twofold. First, we have proposed a data-aware clustering procedure that groups the nodes into spatially homogeneous clusters.

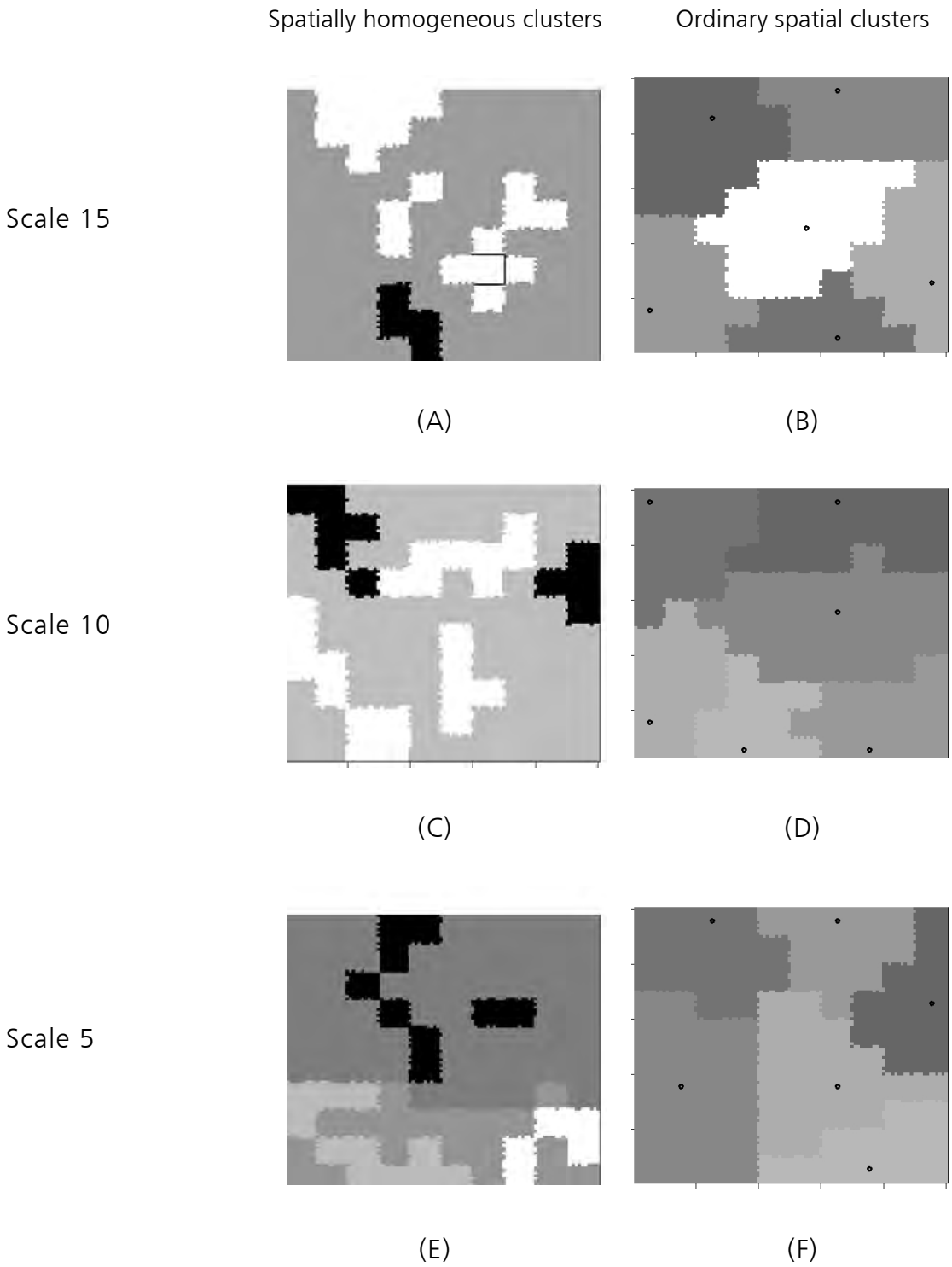
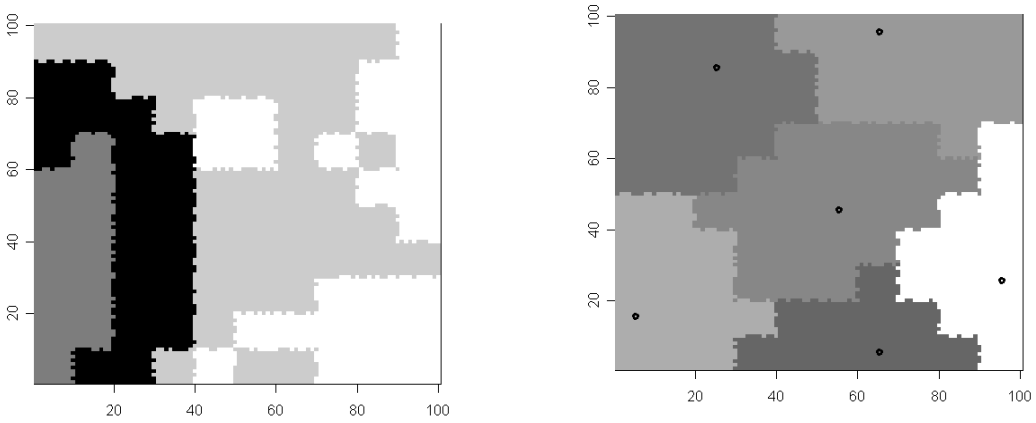


Figure 2.7 - Network data based on the original data in Figure 2.3.
Spatial homogeneous clustering (left column) and ordinary spatial clustering (right column).



Spatially homogeneous clusters Ordinary spatial clusters

Figure 2.8 - Network data based on the original data in Figure 2.4 according the spatial homogeneous clustering and the ordinary spatial clustering.

Second, we have compared our clustering proposal to the usual clustering procedure of cluster-based protocols. We have shown that spatially homogeneous clusters were able to produce data summaries with a higher statistical quality, improving the posterior statistical analysis. In addition, they get better extreme zones identification.

2.7 An updating note

In this chapter, we have considered a sensor network regularly distributed over the study field. Frery *et al*. (2008) have extended our work by evaluating the effect of the distribution patterns for the sensors in the statistical quality of the aggregation estimates. They have used point process to simulate independence, attraction and repulsion of nodes.

Their experiments have shown that the more repulsive the point process, the better the estimates. Similarly to our findings, they have observed that aggregating data according spatially homogeneous clusters produces better estimates of the real data than using the ordinary spatial clusters.

3 DISTRIBUTED DATA-AWARE REPRESENTATIVE CLUSTERING FOR GEOSENSOR NETWORKS DATA COLLECTION♦

Geosensor networks comprise small electro-mechanical devices that sample spatio-temporal fields, collecting data and sending them to a remote base station by wireless communication (NITTEL and STEFANIDIS, 2005). These powerful instruments promise to revolutionize the environmental data collection, such as the monitoring of the wildlife or dangerous environments.

The main goal of a geosensor network is to keep the network's database updated while saving the limited nodes' energy as much as possible.

Since the wireless communication is the main consumer of the nodes' energy, an alternative to reduce the energy consumption is to limit the nodes communication to a local neighborhood, building clusters of nodes. Then, clustering algorithms have gained an important role on the energy-efficient data collection in geosensor networks. They are the basis for the cluster-based data routing protocols (for example, Heinzelman *et al.* (2000) and its variations) and some schemes of spatial and spatio-temporal data suppression (for example, Kotidis (2005) and Tulone and Madden (2006), respectively).

A cluster-based data routing protocol groups the neighboring nodes around a cluster head, which aggregates the data of the cluster members and sends the

♦ This chapter is an extended version of the work in REIS, I. A.; CÂMARA, G.; ASSUNÇÃO, R. M.; MONTEIRO, A. Distributed Data-Aware Representative Clustering for Geosensor Networks Data Collection. In: Brazilian Workshop on Real-Time and Embedded Systems (WRT 2008), 10., Rio de Janeiro, RJ, Brazil. **Proceedings.** Rio de Janeiro: SBC, 2008. p. 77 -- 84. 1 CD-ROM.

summary to the base station. In addition to localize the nodes communication, this strategy reduces the data volume traveling through the network.

A scheme for spatio-temporal data suppression uses the temporal correlation among the readings of a same node and the spatial correlation among the observations of neighboring nodes to infer the expected behavior for the nodes' data. The base station and the nodes agree on this behavior. The nodes send data to the base station only if these data differ from their expected behavior. To deal with the spatial part of the suppression scheme, an alternative is to cluster the nodes around a head node (for example, Kotidis (2005) and Tulone and Madden (2006), respectively).

To meet the energy constraints of a geosensor network, a clustering algorithm must group the nodes using only localized messages (*distributed clustering*) (BASAGNI, 1999). The distributed clustering algorithms for sensor networks can be divided into two categories: *ordinary clustering* (Basagni (1999) and Heinzelman *et al.* (2002), for example) and *data-aware clustering* (Kotidis (2005) and Tulone and Madden (2006), for example). The difference between ordinary and data-aware clustering proposals is on the definition of the nodes' neighborhood (REIS *et al.*, 2007). Ordinary clustering only considers the geographical proximity to define the nodes' neighbors, whereas data-aware clustering constrains this definition, considering also the similarity among the data the nodes sense.

Distributed clustering algorithms have two main tasks: to choose the head nodes and to associate the neighboring non-head nodes to the chosen heads. In the data-aware proposals, a non-head node joins the most similar neighbor head. If there is not a head in the neighborhood of a non-head node, it remains alone (a solitary node). The differences between the data-aware proposals are in the first task (heads choice). Despite of adopting different methods to choose the heads, the usual proposals have the same goal: to find a head node to represent each

associated node *individually* (a *representative node*) (KOTIDIS, 2005), acting as a *nodes' representative* during the data collection.

We look for a distributed data-aware clustering proposal that produces *clusters' representatives*, that is, head nodes that are the result of an *agreement* among neighboring nodes. This agreement considers the interest of all participating nodes. By the interest of a node, we mean "to join its most similar neighbor". In the current proposals, the chosen head cannot be considered a cluster's representative, since the head choice considers the interests of the nodes individually. The nodes in these clusters do not have to be similar to each other.

We believe a cluster built around a cluster's representative produces *more homogeneous* data than a cluster built around a nodes' representative. As a result, a cluster's representative calculates data summaries that are better estimates for the data of its associated nodes (REIS *et al.*, 2007).

The main goal of this chapter is to present two distributed data-aware clustering algorithms that build clusters around clusters' representatives: the *Distributed Data-Aware Representative Clustering* (DARC) and a data-aware version for the DCA (*Distributed Clustering Algorithm*) (BASAGNI, 1999), DA-DCA. In addition, we evaluate our hypothesis on the homogeneity of these clusters.

DARC algorithm promotes a "head election" among neighboring nodes. They exchange information about their most similar neighbor. Then, a node chooses as its head the most often choice among its neighbors, including its own choice. This "election" is the result of the agreement among neighboring nodes and provides a cluster's representative. DA-DCA proposal adapts the neighborhood definition and the heads choice of the original DCA to consider the nodes' data.

Our primary motivation to propose a novel distributed clustering algorithm is to provide the support for the spatial part of a scheme for spatio-temporal data suppression. In this scheme, the head uses cluster summaries to estimate the data its cluster members suppress. Then, we need to build more homogeneous clusters. Moreover, our proposal can provide a method to deal with failure issues inherent to data suppression schemes (SILBERSTEIN *et al.*, 2007a). Since the resultant clusters are homogeneous, the data of a node that fails to deliver its message can be better estimated using the data of a non-failing node in its cluster.

The remainder of this chapter is organized as follows. Section 3.1 describes the related work and section 3.2 presents DARC and DA-DCA proposals. Section 3.3 describes the simulated experiments to evaluate the proposals and presents their main results. Finally, section 3.4 draws some concluding remarks.

3.1 Related Work

One of the first proposals to build clusters of sensor nodes in a localized fashion has been the *Distributed Clustering Algorithm* (DCA) (BASAGNI, 1999). In DCA proposal, each node has a weight (for example, its energy level or its speed, in case of mobile nodes). A node also knows the weights of its neighbors and chooses the one, including itself, that has the biggest weight as its head. If a node chooses itself as a head, it broadcasts a message communicating its status and waits for the joining messages of the other nodes. Since DCA proposal does not involves the sensed data in the clusters building, it is classified as *ordinary spatial clustering* (REIS *et al.*, 2007). Our data-aware version for the clusters building algorithm of the DCA proposal, DA-DCA algorithm, constrains the neighborhood definition of the DCA and adapts the weights used in the heads choice.

Heinzelman *et al.* (2000) have proposed a simpler distributed clustering algorithm as part of a cluster-based data routing protocol. In the LEACH's clustering algorithm, each node "elects" itself as a cluster head according to a user-defined

probability. The chosen nodes broadcast a message communicating their head status. A non-head node listens to the heads' messages and chooses the nearest one as its head node. After associating itself to a cluster head, the non-head node just sends data to its cluster head, which aggregates the data of its cluster's members and sends the summary to the base station. As in DCA, the clustering procedure of LEACH only considers the geographical proximity of the nodes in its neighborhood definition. Then, it is also classified as ordinary spatial clustering. We consider the LEACH's algorithm as one of the simplest and least costly proposals for distributed clustering. Then, we use the results of the LEACH's clustering algorithm as a basis for comparison.

Our work also relates to the data-aware clustering proposals of Kotidis (2005) and Tulone & Madden (2006). Kotidis (2005) has proposed an algorithm to select a small set of *representative nodes* (a *snapshot*) as part of a scheme of spatial data suppression to answer queries. Nodes monitor their neighbors' data messages and estimate the coefficients of a linear regression model to predict their neighbor's data. To define their neighborhood, nodes broadcast their sensed values and listen to the broadcast of their neighbors. Using its neighbors' data, a node estimates a simple linear regression model for each neighbor. Using the estimated models, a node predicts their neighbors' data and compares them with the received data. If the predicted and the real data of a neighbor differ by less than a threshold θ_{SNAP}, this neighbor enters to the candidates list of the node. After completing their candidates lists, nodes broadcast them and listen to their neighbors' lists. A node chooses as its representative the neighbor node with the longest candidates list. Once the representative nodes are chosen, only they answer the queries. In the snapshot maintenance, non-representative nodes continuously monitor their representatives' data. Whenever they differ from the data the non-representative predicts, the node looks for another representative, repeating the initial steps.

Representative nodes also appear in the spatial version of PAQ, a scheme for temporal data suppression using time series models (TULONE and MADDEN, 2006). PAQ's algorithm simplifies the Kotidis' proposal. It evaluates the similarity of two nodes' data only comparing their difference to a similarity threshold θ_{PAQ}. Furthermore, nodes do not exchange their list of similar neighbors to choose their representatives. Once the node has its list of similar neighbors, it includes its own ID in this list and chooses as its representative (the head) that node with the lowest ID. If a node is a head, it broadcasts a message communicating its status. A non-head receives a head message and checks if the head ID belongs to its list of heads. If so, it sends a joining request to the head node. Otherwise, it keeps listening to the heads' messages until the joining period ends. After that, if a node did not receive messages of the heads in its list, it remains alone and looks for a head in the next time period (clusters maintenance). As in the Snapshot algorithm, only the representative nodes (heads) send data to the base station.

Our DARC algorithm has been inspired in PAQ's and Kotidis' algorithms to build snapshots. We have adopted the simple evaluation of the nodes' similarity of PAQ and adapted the "neighbors' conversation" in Kotidis' proposal. In DARC algorithm, neighboring nodes exchange information about their most similar neighbor. Then, a node chooses as its head the most often choice among its neighbors. This transforms the "neighbors' conversation" into "neighbors voting" and the heads choice in a real "heads election". In addition, we propose an adjusting time period, which gives to the head nodes without a cluster (solitary nodes) a last chance to get a cluster at the end of the nodes association phase.

Differently from Snapshot and PAQ's clustering algorithms, DARC and DA-DCA algorithms promote the heads' rotation in their clusters maintenance phase. This procedure distributes the costly tasks of being a head among all the nodes in the network.

To build the initial clusters, PAQ's algorithm spends up to two local messages per node and Kotidis' algorithm spends up to six messages, whereas DARC and DA-DCA algorithms spend up to four messages per node. To maintain the clusters, DARC and DA-DCA spend from zero to three messages per node, whereas PAQ's and Kotidis'algorithms spend up to two and six messages per node, respectively.

The goal of the representatives in Kotidis' and PAQ's proposals is to represent each associated node *individually*. To represent a single associated node, the representative does not need the data of the other associated nodes. In our proposals, the representative nodes (the heads) are *clusters' representatives*. They compute the *average* of the data of *all* nodes in the cluster and the resulting value *estimates* these nodes' data. This estimation procedure allows for spatial suppression as well as for local monitoring of the cluster area.

Reis *et al.* (2007) have concluded that *spatially homogeneous clusters* (SHC) produce data summaries that are better estimates of the summarized data if compared with the summaries based on ordinary clusters. SHC are the result of a partition of the sensor field that has maximum internal homogeneity regarding the values of monitored variable. To get such clusters, the authors have used the SKATER (*Spatial 'K'luster Analysis by Tree Edge Removal*). SKATER is a centralized data-aware clustering procedure, since it is necessary to know the values of all nodes in the network to maximize the clusters internal homogeneity. In the sensor network context, SKATER can be considered as an information fusion system (NAKAMURA *et al.*, 2007). Regarding the communication costs, a centralized fusion system may outperform a distributed one (TENNEY and SANDELL JR., 1981). Therefore, we use SKATER's clusters to have the lower bounds for the evaluation measures: the internal homogeneity of the cluster and the prediction error of the cluster average.

3.2 Our Proposals for Distributed Data-Aware Clustering

From now on, we reserve the term "similar neighboring nodes" or just "neighbors" to denote those geographical neighbors that collect similar data. As "geographical neighbors", we mean those nodes that can communicate to each other.

In this chapter, we define the similarity of two values v_i and v_q as $d_i = |v_i - v_q|$. The value v_i is considered to be similar to v_q if $d_i \leqslant \theta_{(S)i}$, where $\theta_{(S)i}$ is a similarity threshold. In the DARC and DA-DCA algorithms, $\theta_{(S)i} = \theta \times MAD_i$, where MAD_i is the mean absolute deviation of the measures of the node that evaluates the similarity between its data and its neighbor's data. MAD is a measure of data variability and represents the typical data deviation. MAD is less costly the usual standard deviation, since it does not require the square and square root operations[12]. The definition for $\theta_{(S)i}$ allows for standardizing the difference between two sensed data. This makes easier the choice of the similarity parameter θ, since θ represents the maximum number of typical data deviations that separates two similar values. For instance, we consider as similar two sensed data apart from each other at most four typical deviations, that is, $\theta=4$. The value of MAD_i is estimated during a learning phase.

3.2.1 Distributed Data-Aware Representative Clustering (DARC) Algorithm

The main idea of DARC algorithm is to get an agreement among neighboring nodes to choose one of them as their cluster head. This agreement results from the exchange of local messages among neighboring nodes. Figure 3.1 describes

[12] In REIS *et al*. (2008), we have used the standard deviation instead of MAD. In this chapter, we have acknowledged the suggestion of Prof. Alejandro Frery and adopted MAD, since it is a less costly function than the standard deviation.

the types of messages that nodes receive or send during the two phases of the DARC algorithm: clusters building and maintenance. In the first phase, non-head nodes send three local messages and head nodes have to send one more local message. In the clusters maintenance, nodes send from zero to three local messages. A typical message has the format *<message head, receiver's ID, sender's ID, message content (c)>.*

1) <*hello,* ID_i, v_i> : from node ID_i to its geographical neighbors. This message contains the measured value v_i. If v_i is missing (v_i=MS), this is a message from a head node.

2) <*near,* $IDvn_i$> : from node ID_i to geographical neighbors. This message contains the ID of the nearest neighbor of node i, $IDvn_i$.

3) <*join,* ID_{CH}, ID_i> : from node ID_i to head ID_{CH}. This message contains a join request.

4) <*head,* ID_i> : from head ID_i to its geographical neighbors. This message contains the head status of node ID_i.

5) <*abandon,*ID_{CH},ID_i> : from node ID_i to head ID_{CH}. This message contains an abandoning notification.

6) <*decline,*ID_i> : from head ID_i to geographical neighbors. This message contains a declining notification.

7) <*data,* ID_i,ID_q, v_i> : from node ID_q to node ID_i. This message contains a data value.

8) <*headdata,* ID_i,avg_i> : from head ID_i to the members of its cluster. This message contains the average value of the cluster.

Figure 3.1 - Types of messages the nodes exchange during the network operation.

The clusters' building begins after the learning phase. Figure 3.2 presents a pseudo-code describing the steps of the clusters' building and Figure A.1 in the Appendix presents a description of DARC as a distributed algorithm. The total time period for clusters building is divided into four time periods:

Phase A - Building the initial clusters

A.1) - Node ID_i broadcasts a message $<hello,\ ID_i\ , c=v_i>$ to all nodes within its radio range (neighbors candidates).

A.2) - Node ID_i receives messages from its neighbors candidates, $<hello, ID_q, c=v_q>, q \neq i$. If $|v_i - v_q| \leqslant (\theta \times MAD_i)$, $\{N_i\} \leftarrow \{N_i\} \cup ID_q$.

A.3) - Node ID_i chooses its neighbor with the smallest value for $|v_i - v_q|$ as its head candidate, ID_q, and

- broadcasts a message with its candidate, $<near, c=ID_q>$, to all its neighbors;
- initializes its list of possible heads, $\{CH^{(list)}_i\} \leftarrow ID_q$.

A.4) - Node ID_i receives messages from its neighbors with their head candidates, $<near, c=ID_q>$, $q \neq i$.

- if $ID_q \in \{N_i\}$, $\{CH^{(list)}_i\} \leftarrow \{CH^{(list)}_i\} \cup ID_q$.

A.5) - Node ID_i chooses the most often node(s) in $\{CH^{(list)}_i\}$ and excludes the other nodes from the list $\{CH^{(list)}_i\}$;

- if none of the nodes in $\{CH^{(list)}_i\}$ is the most often, $\{CH^{(list)}_i\} \leftarrow \{N_i\}$.

A.6) - If $ID_i \in \{CH^{(list)}_i\}$, node ID_i

- sets $CH_i \leftarrow ID_i$;
- broadcasts a message with its head status, $<head,\ c=ID_i>$;
- initializes its counter as a head node, $CH.count_i \leftarrow 1$;
- waits for joining requests (step A.8).

Figure 3.2 - Cluster building phase of the DARC algorithm. (*to be continued*)

A.7) - If $CH_i \neq ID_i$ (node ID_i **is not** a head):

A.7.1) - It receives messages from head candidates ID_h:

A.7.1.1) - if $ID_h \in \{CH^{(list)}_i\}$, $\{CH^{(cand)}_i\} \leftarrow \{CH^{(cand)}_i\} \cup ID_h$;

A.7.1.2) - if ($ID_h \notin \{CH^{(list)}_i\}$ AND $ID_h \in \{N_i\}$), $\{CH^{(wait)}_i\} \leftarrow \{CH^{(wait)}_i\} \cup ID_h$.

A.7.2) - After the association time (Δ_{TA}), node ID_i

- chooses the nearest member of $\{CH^{(cand)}_i\}$ as its head and
- sets $CH_i \leftarrow ID_{CH}$, the ID of the chosen head node :
  - If $\{CH^{(cand)}_i\}=\varnothing$, node ID_i chooses the nearest member of $\{CH^{(wait)}_i\}$ as its head ;
  - If $\{CH^{(wait)}_i\}=\varnothing$, node ID_i sets $CH_i \leftarrow ID_i$ and remains alone (a solitary node).
- sends a join message to the node in CH_i, $<join,ID_{CH},ID_i>$, (except when $CH_i=ID_i$).

A.8) If $CH_i = ID_i$ (node ID_i **is** a head):

A.8.1) - It receives the join messages, $<join,ID_i,ID_q>$:

- if $ID_q \in \{N_i\}$,
  - sends message $<data,ID_i\ ID_q,c=\varnothing>$ to node ID_q;
  - sets $\{CL_i\} \leftarrow \{CL_i\} \cup ID_q$.

A.8.2) - It receives the messages $<head,ID_h>$ from other heads and sets $\{CH^{(wait)}_i\} \leftarrow \{CH^{(wait)}_i\} \cup ID_h$;

A.8.3) - At the end of the joining time (Δ_{TJ}), if $\{CL_i\} = \varnothing$:

- node ID_i sends a join message, $<join,\ ID_q,\ ID_i>$, to the first neighbor head of its list $\{CH^{(wait)}_i\}$, ID_q;
- if $\{CH^{(wait)}_i\} = \varnothing$, node ID_i remains alone (a solitary node).

A.8.4) - At the end of the adjusting time (Δ_{TAD}), if the head ID_i did not receive any join message, it remains alone (a solitary node).

Figure 3.2 - Cluster building phase of the DARC algorithm (*conclusion*)

1) *Talking time* (Δ_{TT}): nodes exchange messages with their sensed values (steps A.1 to A.4). During the talking time, there is a time period to constrain the neighborhood (Δ_{TN}), where $\Delta_{TN} < \Delta_{TT}$.
2) *Association time* (Δ_{TA}): nodes choose their heads and decide their status (head or non-head). Heads broadcast their status and all nodes listen to the messages (steps A.5, A.6, A.7.1).
3) *Joining time* (Δ_{TJ}): non-head nodes send joining messages and heads listen to them (steps A.7.2, A.8.1 and A.8.2).
4) *Adjusting time* (Δ_{TAD}): Head nodes without a cluster have a last opportunity to join other heads executing steps A.8.3 and A.8.4. All head nodes keep listening to the joining messages, while non-head nodes switch their radios to the sleep mode until the end of the adjusting time.

At the end of the initial clustering, nodes have one of three conditions: head, non-head or solitary node. A head node has at least one associated non-head node. A non-head belongs to one cluster and a solitary node does not belong to any cluster.

We explain DARC proposal using the example in the Figure 3.3A to 3.3E. Figure 3.3A presents a sensor network, the nodes' ID (inside the circles) and the value they sense (v) in a time period *t*. We represent the spatial variation of the process being monitored painting the nodes according to the value sensed: yellow nodes for $v<5$ and red nodes for $v\geq5$. The edges represent the radio links among the nodes. The similarity threshold is $\theta=4$ and $s_i = 1$, for $i = 1,2,3,4,5,6,7,8,9$.

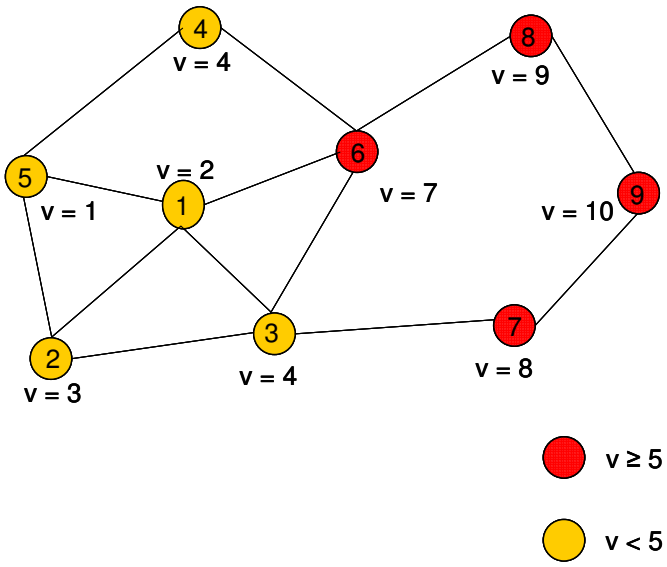
From steps A.1 to A.3, each node discovers its geographical neighbors, refines this neighborhood discarding those nodes with non-similar data (Figure 3.3B) and chooses the most similar neighbor to be its head candidate (neighbors inside of the rectangles). Note the node ID_6 is within the radio range of node ID_1, but it is

not part of the neighbors' list of ID_1. This is because their data are not similar, since $|v_1 - v_6| > \theta$. In step A.4, nodes communicate the chosen heads to the neighbors and receive their choices. Then, the nodes build the list $\{CH^{(list)}\}$, which contains their own chosen heads and the choices of their neighbors that belong to the list $\{N_i\}$ (Figure 3.3C). The $\{CH^{(list)}\}$ of node ID_4, for example, has only its chosen head (ID_5), since the heads chosen by its neighbors (ID_1 and ID_8) does not belong to its neighbors list.

The agreement among nodes' choices occurs at step A.5, when each node refines its list $\{CH^{(list)}\}$ keeping only the most "popular" node(s) among the chosen heads (Figure 3.3C). If there is not a most often node in the original $\{CH^{(list)}\}$ or if the original $\{CH^{(list)}\}$ has only node's choice (for example, nodes ID_4, ID_6 and ID_7), the refined $\{CH^{(list)}\}$ is the node's choice. If the node's ID belongs to the its refined $\{CH^{(list)}\}$ (step A.6), it sets its status to head, broadcasts its status and goes to step A.8. In Figure 3.3C, nodes ID_1, ID_2, ID_5, ID_8 and ID_9 set their status as heads. The non-head nodes (ID_3, ID_4, ID_6 and ID_7) follow the instructions in the step A.7. They listen to the heads' messages and build two lists of nodes: $\{CH^{(cand)}\}$, which has heads belonging to the refined $\{CH^{(list)}\}$ list (inside of the ellipses) and $\{CH^{(wait)}\}$, which has the other neighbor heads (Figure 3.3D) . The head nodes also build the list $CH^{(wait)}$. At the end of clustering period, non-head nodes choose the most similar head in the $CH^{(cand)}$. If $CH^{(cand)}$ is empty, they choose the first head in the waiting list $CH^{(wait)}$. If $CH^{(wait)}$ is empty, they become solitary nodes and keep this status until the maintenance phase.

The head nodes follow the step A.8. They receive the joining requests of the non-head nodes and the announcements of neighbor heads, keeping the ID of the head nodes in the waiting list $CH^{(wait)}$. If a head node does not receive any joining request, it uses the list $CH^{(wait)}$ to join another head in its neighborhood. In the example of Figure 3.3, node ID_1 does not receive any joining message, since it is not chosen as head by any node. Then, in the adjusting time period, it uses its

$CH^{(wait)}$ to join the node ID_5. If the $CH^{(wait)}$ is empty, the node changes its status to a solitary node and keeps this status until the maintenance phase, when it tries to join a cluster.



(A) Nodes' neighborhood and sensed data *v*

$\{N_1\} = \{\boxed{ID_2}, ID_3, \boxed{ID_5}\}$
$\{N_2\} = \{\boxed{ID_1}, ID_3, ID_5\}$
$\{N_3\} = \{ ID_1, \boxed{ID_2}, ID_6, ID_7\}$
$\{N_4\} = \{\boxed{ID_5}, ID_6\}$
$\{N_5\} = \{\boxed{ID_1}, ID_2, ID_4\}$
$\{N_6\} = \{ ID_3, ID_4, \boxed{ID_8}\}$
$\{N_7\} = \{ ID_3, \boxed{ID_9}\}$
$\{N_8\} = \{ ID_6, \boxed{ID_9}\}$
$\{N_9\} = \{ ID_7, \boxed{ID_8}\}$

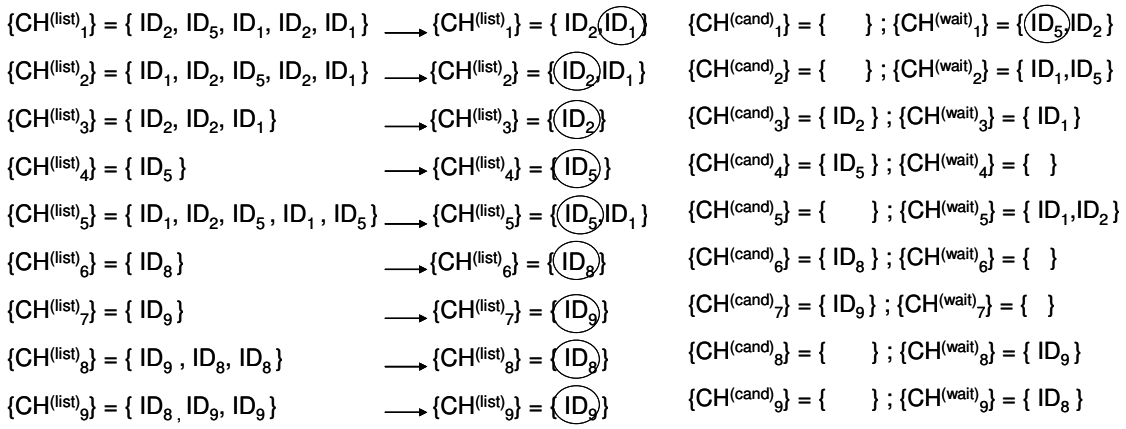
(B) Neighbors lists and the nearest neighbors, inside of the rectangles

Figure 3.3 - Example for explaining DARC proposal (*to be continued*).

Figure 3.3E presents the resulting clusters. The initial nine nodes are grouped into four clusters and there is no solitary node. The strategy of keeping a waiting list and having an adjusting time period avoids a large number of solitary nodes. Although it is a simple example, it is worth to note that the resulting DARC's clusters have preserved the original spatial division into small and large sensed values (yellow and red nodes).

If we apply the PAQ's grouping algorithm (TULONE and MADDEN, 2006) to the network of Figure 3.3A, we get only one cluster ($\{ID_1, ID_2, ID_3, ID_5\}$) and five

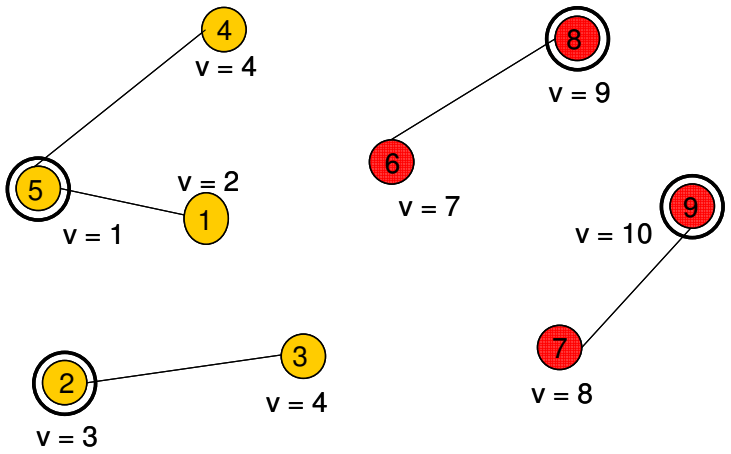
solitary nodes. Since PAQ does not have an adjusting time period, it produces a larger number of solitary nodes.



(C) Lists of head candidates and chosen heads (inside of the ellipses

(D) Candidates and waiting lists

Figure 3.3 (*continuation*) - Example for explaining DARC proposal (*to be continued*)



(E) Final clusters (heads nodes are circled)

Figure 3.3 - Example for explaining DARC proposal (*conclusion*).

3.2.2 DA-DCA (Data-aware Distributed Clustering Algorithm)

The differences between DA-DCA and DCA (BASAGNI, 1999) are the neighborhood and the weights definitions, since these definitions in DA-DCA consider the sensed data. The weight w_i of a node ID_i is a function of its data and its neighbors' data. The weight w_i is the average of the absolute differences $|v_q - v_i|$, where $ID_q \in \{N_i\}$ and $\{N_i\}$ is the list of neighbors of the node ID_i. The smaller the value of w_i, the more similar to its neighbors the node ID_i is. Then, a node chooses among its neighbors, including itself, the one that has the smallest weight to be its head. DA-DCA does not require a similarity threshold as in DARC, SNAP and PAQ, which can be interesting for the network's user.

We use the example of the Figure 3.3A to present the DA-DCA. The weights of the nodes ID_1 to ID_9 are 1.33, 1.33, 2.50, 3.00, 2.00, 2.67, 3.00, 1.50 and 1.50, respectively. Then, according to neighbors list in the Figure 3.3B, the choices of nodes ID_1 to ID_9 are ID_1, ID_2, ID_2, ID_5, ID_1, ID_8, ID_9, ID_8 and ID_9, respectively. Since the nodes ID_1 and ID_2 have the same weight, the nodes ID_3 and ID_5 have to break the tie choosing the most similar head candidate. The nodes ID_1, ID_2, ID_8 and ID_9 are the chosen heads. The final clusters are $\{ID_1,ID_5\}$, $\{ID_2,ID_3\}$, $\{ID_4\}$, $\{ID_8,ID_6\}$ and $\{ID_9,ID_7\}$. Except by the solitary node ID_4, the resulting clusters are equal to DARC's clusters.

3.2.3 Clusters maintenance

Once DARC or DA-DCA algorithms builds the initial clusters, the nodes start the clusters maintenance phase. The goal of this phase is to adapt the initial clusters to data dynamics and to allow for rotating the heads.

To avoid a long period as a head, the nodes maintain two counters: a *CH.count* and a *rest.count*. The first one stores the number of sequential time periods the node is a head and is initialized at the beginning of each period as a head. The

second counter stores the number of sequential time periods the node is not a head and is initialized when a head node turns its status to non-head. The node updates its counters *CH.count* or *rest.count* whenever it acts as a head or a non-head, respectively. At each sampling period, the head node checks its counter and decides to remain as a head or not. Solitary nodes use their "rest period" counter to accept or not a joining request of another node. The maximum size of the sequential period as a head (T_{asCH}) is a user-defined parameter as well as the maximum proportion of the sampling time periods a node can be a head (P_{asCH}). The size of the rest period (T_{rest}) depends on T_{asCH} and on P_{asCH}: $T_{rest} = T_{asCH} \times (1/ P_{asCH} - 1)$. For instance, if we set a node can be a head for up to 25% of the sampling time periods and for up to 10 sequential time periods, the rest period is $T_{rest} = 10 \times (1/ 0.25 - 1) = 30$ time periods.

The clusters maintenance depends on a scheme for temporal data suppression adopted by the nodes to decide when to send data to their heads. A simple alternative for temporal data suppression is to send v_t, the sensed data at time period *t*, only if $| v_t - v_L | > \varepsilon$, where v_L is the last data sent to the head and ε, $\varepsilon > 0$, is a suppression threshold defined by the user. In this section, we describe the clusters maintenance without concerning about any particular temporal data suppression scheme.

The clusters maintenance is divided into three sequential time periods: *sampling*, *evaluation* and *searching*. During the sampling time, non-heads and solitary nodes sense the environment and decide to send or not their data according to the adopted scheme for temporal suppression. The heads sense the environment and waits for data messages from their associated nodes.

During the next time period (evaluation), the cluster head evaluates changes in cluster homogeneity. For those members that decided to send data in the sampling period, the head evaluates the impact of their changes in the cluster

homogeneity. To do this, the head calculates a measure of the homogeneity of the cluster's members.

The homogeneity of a dataset is usually measured by a coefficient between a measure for data dispersion and a measure for the typical data value. The smaller the data dispersion in relation to the typical data value, the more homogeneous the data set is. There are some alternatives for measuring data dispersion such as the the standard deviation, the median absolute deviation[13] and the mean absolute deviation[14]. The correspondent measures for the typical data value are the average, the median and the average, respectively. The median and the mean absolute deviation have the advantage of being less costly than the standard deviation, since they do not use the functions square and square root. Then, suppose we adopt one of these alternatives to measure the clusters' homogeneity. We refer to this choice as H.

To evaluate the impact of their changes in the cluster homogeneity, the head node calculates H using the old values of the cluster's members (H_{old}) and compares it with the H value based on the new values (H_{new}). If the ratio H_{new}/H_{old} is greater than a threshold δ_H ($\delta_H > 1$), the head considers the impact of the node in the cluster homogeneity as large and excludes the node of the cluster by sending it a declining message. In fact, this evaluation procedure must be iterative, similarly to the *stepwise* procedure to select variables to be part of regression model (DRAPER and SMITH, 1998). That is because the head has to evaluate all possible

[13] The Median Absolute Deviation of values v_i, i=1,2,..,N, is defined as $median_{i=1,2,...,N}\left(\left|v_i - \tilde{v}\right|\right)$, onde $\tilde{v} = median_{i=1,2,...,N}(v_i)$.

[14] The Mean Absolute Deviation of values v_i, i=1,2,..,N, is defined as $\sum_{i=1}^{N}\left|v_i - \overline{v}\right| \Big/ N$, onde $\overline{v} = \sum_{i=1}^{N} v_i \Big/ N$.

combinations of members to define the set of them whose preserve the internal homogeneity of the last time period, that is, $(H_{new}/H_{old}) < \delta_H$. For example, let us consider the cluster $\{ID_1,ID_2\ ID_3,ID_4,ID_5\}$. Suppose nodes ID_1, ID_3 and ID_4 have sent their values to the head. So, we have to evaluate the following subsets : $\{ID_2,ID_5,ID_1\}$, $\{ID_2,ID_5,ID_3\}$, $\{ID_2,ID_5,ID_4\}$, $\{ID_2,ID_5,ID_1,ID_3\}$, $\{ID_2,ID_5,ID_1,ID_4\}$, $\{ID_2,ID_5,ID_3,ID_4\}$. Since nodes ID_2 and ID_5 have suppressed their values, they are in all subsets. Indeed, we are looking for the largest subset of nodes that preserve the old cluster homogeneity. Then, the head starts the procedure evaluating the largest subsets. In the worst case, the head wil have to evaluate $(2^m\text{-}2)$ subsets, where $m \geq 1$, is number of nodes that have sent their values to the head[15]. In the previous examples, $m = 3$ and the maximum number of subsets to evaluate will be $2^3 - 2 = 6$ subsets, which are listed above. If the head does not find a subset of members that preserve the old cluster homogeneity, it dissolves the cluster, sends a declining message to its members and becomes a solitary node.

Once having the subset of nodes which preserves the cluster homogeneity, the head computes the cluster average. After evaluating the changes in the value of its cluster average using the adopted temporal scheme, the head checks its time period as a cluster head and decide to dissolve or not the cluster. During the evaluation time period, non-head nodes wait for messages from the head. If a non-head node receives a declining message from its head, it becomes a new solitary node. During the evaluation time period, the old solitary nodes turn their radios to the sleep mode, since DARC and DA-DCA algorithms reserve this period to the communication between the cluster head and its associated nodes.

[15] We have to evaluate $\sum_{k=1}^{m-1}\binom{m}{k}$ subsets. Using the Binomial Theorem, $\sum_{k=1}^{m-1}\binom{m}{k} = 2^m - 2$.

During the next time period (searching), old and new solitary nodes have the chance to join a cluster or to become a cluster head. The head nodes receive the joining requests of the solitary nodes and evaluate the impact of the new node in the cluster homogeneity. Similarly to the evaluation period, a new node only is accepted in a cluster if its inclusion preserves the cluster homogeneity.

It is worth to note that only cluster heads and solitary nodes are allowed to send messages to base station. In fact, combining DARC or DA-DCA algorithm and a scheme for temporal data suppression would produce a scheme for spatio-temporal data suppression. Since evaluating this kind of scheme is not our goal in this chapter, we will only describe cluster maintenance phase, not carrying out any evaluation of performance in this chapter.

3.3 Assessing the Performance of the Clustering Algorithms

This section presents the main results of the simulated experiments we have carried out to provide a preliminary evaluation of the performance of the clustering algorithms DARC, DA-DCA, SNAP and PAQ.

3.3.1 The experiments

We have simulated datasets according to a Gaussian random field using a grid of 10000 cells (100 x 100), using the same procedure described in Chapter 2 (Section 2.5). As in REIS *et al.* (2007), we refer to these data as *original data*. These datasets are characterized by *zones*, which are groups of cells with similar values. The zones' size relates to the spatial autocorrelation and is controlled by a scale parameter. The higher scale value, the larger the zones size. To each value of the scale parameter (5, 10, 15, 20, 30 and 40), we have simulated 1000 spatial datasets with the same mean (100) and variance (10). To sample the original data, we have deployed a geosensor network with 100 nodes in a regular fashion. As in Chapter 2, each sensor node has an *area of influence*, which is the area

around the node. In our experiments, we have defined the *area of influence of a sensor node* as the set of cells of which the node was the nearest node. The data collected by a sensor node were defined as the average of the values of the cells in its area of influence[16]. We refer to this sample as *geosensor data*.

The nodes have been grouped according to the four clustering algorithms under evaluation: our proposals (DARC and DA-DCA), Kotidis' proposal (SNAP) and PAQ. We have set the radio range equal to 20, which represents the double of the distance between two adjacent nodes in the regular grid. The idea is to localize the nodes communication to save energy. The user sets a short radio range to be used in most time periods, saving the entire range for an emergency, as a long time period without communication with local neighbors, for instance.

The data-aware clustering proposals we consider here define different similarity measures to constrain the geographical neighborhood of the nodes. To make these algorithms comparable, we have adjusted their similarity thresholds. The similarity measure of SNAP is defined as $\left|\hat{v}_{iq} - v_{iq}\right|$, where v_{iq} is the value sensed by the $q\text{-}th$ neighbor of the node ID_i and $\hat{v}_{iq}$ is the value predicted for $q\text{-}th$ neighbor of the node ID_i using the regression model that node ID_i estimates for its $q\text{-}th$ neighbor. The similarity measure of SNAP is the absolute value of the prediction error of a regression model. To build the SNAP's regression models, we have run the learning phase for 100 times periods.

[16] In REIS *et al*. (2008), the data collected by a sensor node were defined as the value of the cell in which the node was placed.

Since DARC uses a statistical property (the mean absolute deviation) of the nodes' data to define its similarity threshold ($\theta_{(S)i} = \theta \times MAD_i$), we have adopted an equivalent procedure to define the similarity thresholds of the other proposals. For PAQ and SNAP, we have set $\theta_{(PAQ)i} = \theta_{(S)i}$ and $\theta_{(SNAP)iq} = \theta_{(S)i}$, respectively[17]. The estimate for MAD_i has been calculated during the SNAP learning phase. To calculate $\theta_{(S)i}$, we have adopted a fixed value for θ ($\theta = 4$), since we are not interested in studying the effect of similarity threshold on the performance of algorithms. Increasing the value of θ will increase the clusters size but decrease the clusters homogeneity.

For each clustering algorithm, we have calculated the number of resulting clusters (n_C) and the number of solitary nodes (n_S). To each cluster *k*, we have calculated the cluster size (n_k), the *Median of the Absolute Value of the Relative Error* of the cluster average ($MARE_k$) and the *Coefficient of Variation* (CV_k), which are defined by the expressions

$$MARE_k = \underset{i=1,\ldots,n_k}{median}\left(\frac{\left|v_{ik} - \mathrm{CM}_k\right|}{v_{ik}}\right) \text{ and} \tag{3.1}$$

$$CV_k = \frac{\mathrm{Sd_k}}{\mathrm{CM}_k}, \tag{3.2}$$

where v_{ik} is the data sensed by the node ID_i of the cluster *k*; CM_k and Sd_k are the average and the standard deviation of the data sensed by the members of the cluster *k*, respectively. Since we have simulated values larger than zero, $v_{ik} \neq 0$, for all *i* and *k*. $MARE_k$ measures the prediction error of cluster average CM_k as an

[17] In REIS *et al.* (2008), we have adopted the standard error of a 95% confidence interval for v_{iq} as the similarity threshold for SNAP.

estimate for the data of the cluster *k* members, whereas CV_k measures the internal homogeneity of the cluster *k*.

To produce the bounds of comparison for $MARE_k$ and CV_k, we have used the clustering algorithms LEACH and SKATER, as we have discussed in section 3.1. LEACH and SKATER should produce the upper and the lower bounds to $MARE_k$, respectively. The same reasoning is valid for CV_k. Therefore, for each dataset, we have used the number of clusters produced for a given clustering proposal to define the number of clusters to be assembled using LEACH and SKATER. Then, we have calculated the *ratio* between the median of the $MARE_k$ values of a given clustering proposal and the median of the $MARE_k$ values of the corresponding LEACH's clusters. Similarly, we have calculated the $MARE_k$ ratios in relation to SKATER's clusters. We have used the same procedure to obtain the CV_k ratios.

3.3.2 The results

To summarize the results for $MARE_k$ values of the evaluated data-aware clustering proposals in the 1000 simulated datasets, we have prepared the Figure 3.4.

The mean values based on DARC's clusters have got the smallest prediction error, in median, in addition to the smallest variability. DA-DCA and SNAP algorithms have got very similar performances, even though they use very different rules to cluster the network's nodes. The simplest clustering proposal, PAQ, has got the poorest performance.

Since LEACH's clustering algorithm assembles ordinary spatial clusters, we expected the clustering proposals evaluated here could improve the LEACH's clusters performance. In other words, we expected their $MARE_k$ and CV_k values were smaller than LEACH's values. On the other hand, we expected they were larger than SKATER's values, since the centralized clustering procedure of SKATER is able to produce more homogeneous partition of the nodes than a distributed procedure.

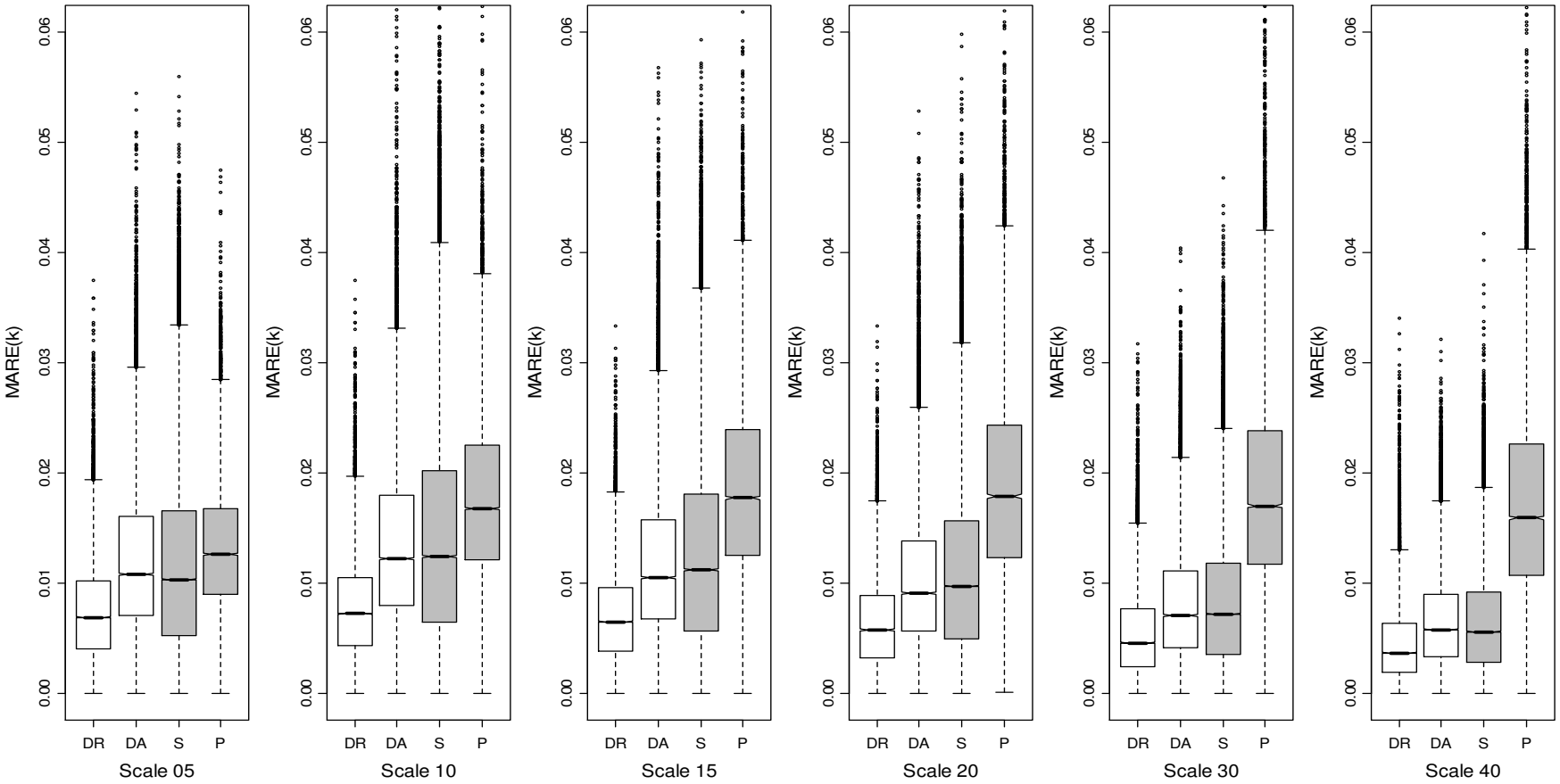


Figure 3.4 - Boxplots for $MARE_k$ values of the evaluated clustering proposals according to the scale parameter.

Legend: DR, DA, S and P stand for DARC, DA-DCA, SNAP and PAQ, respectively.

To compare the evalatued clustering proposals with LEACH and SKATER, we have prepared figures 3.5 to 3.8. Since the results for CV_k are quite similar to the results for $MARE_k$ results, we have opted for using only the figures summarizing $MARE_k$ results to discuss the results for both measures.

Analyzing figures 3.5 to 3.8, we can draw the following conclusions:

(a) DARC algorithm improved the clusters internal homogeneity compared to the LEACH's algorithm (smaller values for CV_k, in median). In addition, the prediction error of the average of DARC's clusters is smaller than the LEACH's clusters error (smaller values for $MARE_k$, in median);

(b) the performance of DARC's clusters is the most similar to the performance of SKATER's clusters ($MARE_k$ and CV_k for DARC's clusters have been the nearest values to values of SKATER's clusters, in median);

(c) except by DARC, the other clustering proposals could not improve the LEACH's ordinary clustering.

As expected, the values for MARE and CV of all clustering proposals decrease as the spatial correlation (zones size) becomes higher, especially if the scale parameter is larger than 15. However, for the SKATER algorithm, these values decrease faster. That is because neither the shape nor the size of SKATER clusters is constrained by the radio range. Furthermore, the geosensor data is able to capture the spatial patterns as the spatial correlation increases and SKATER is very sensitive to this (REIS *et al.*, 2007).

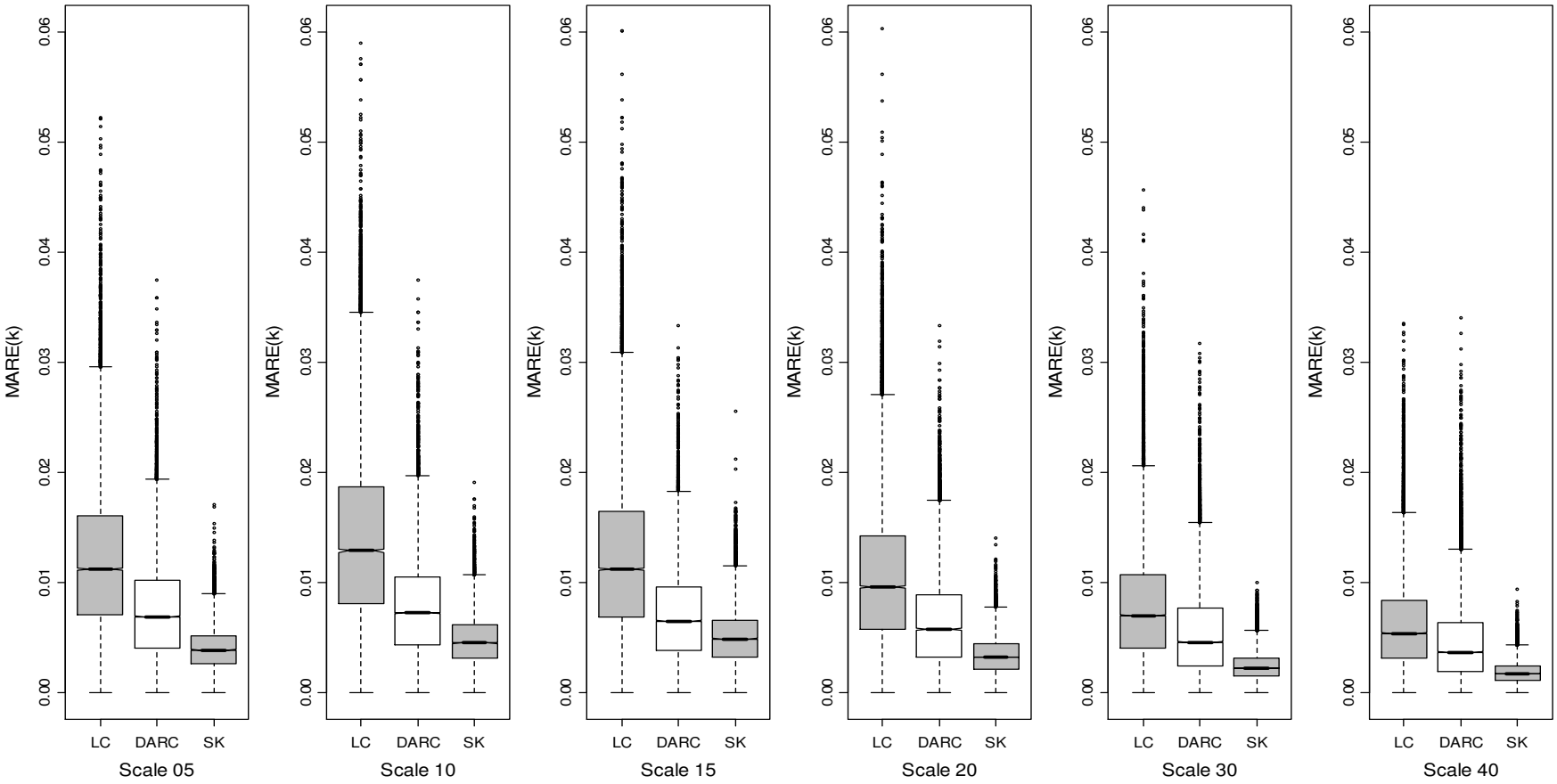


Figure 3.5 - Boxplots for $MARE_{\underline{k}}$ values of DARC, LEACH (LC) and SKATER (SK) according to the scale parameter.

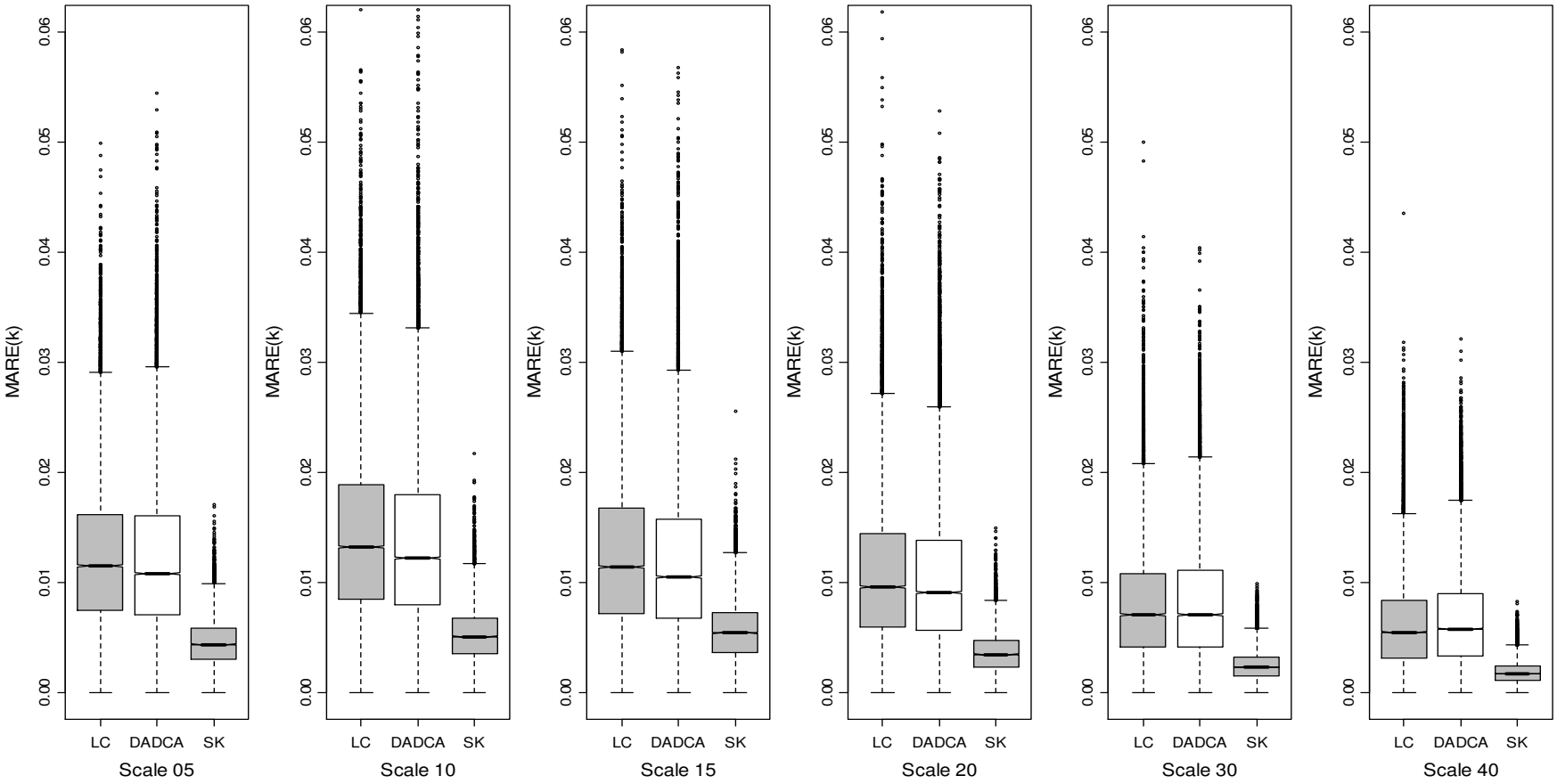


Figure 3.6 - Boxplots for $MARE_k$ values of DA-DCA, LEACH (LC) and SKATER (SK) according to the scale parameter.

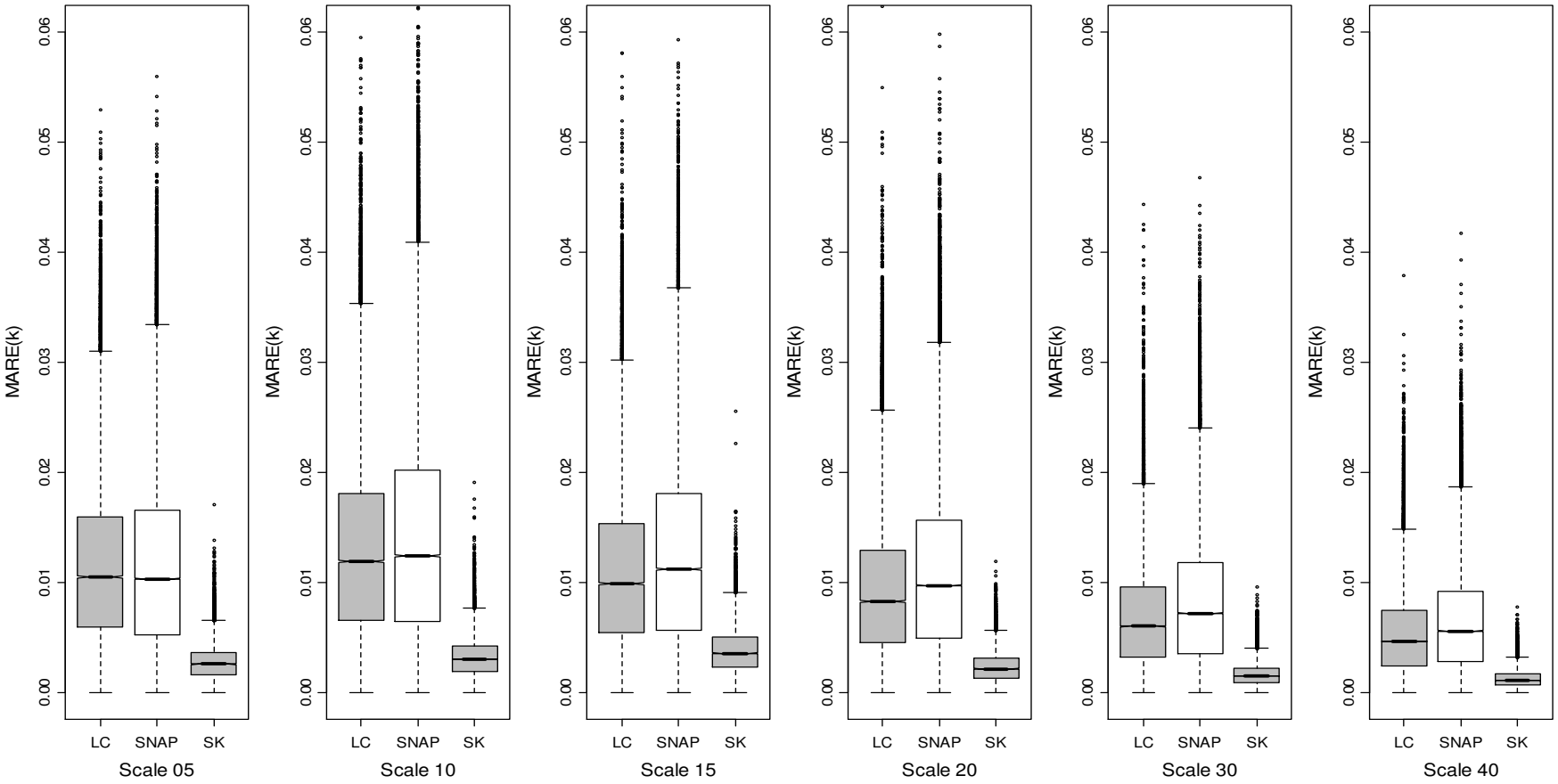


Figure 3.7 - Boxplots for $MARE_k$ values of SNAP, LEACH (LC) and SKATER (SK) according to the scale parameter.

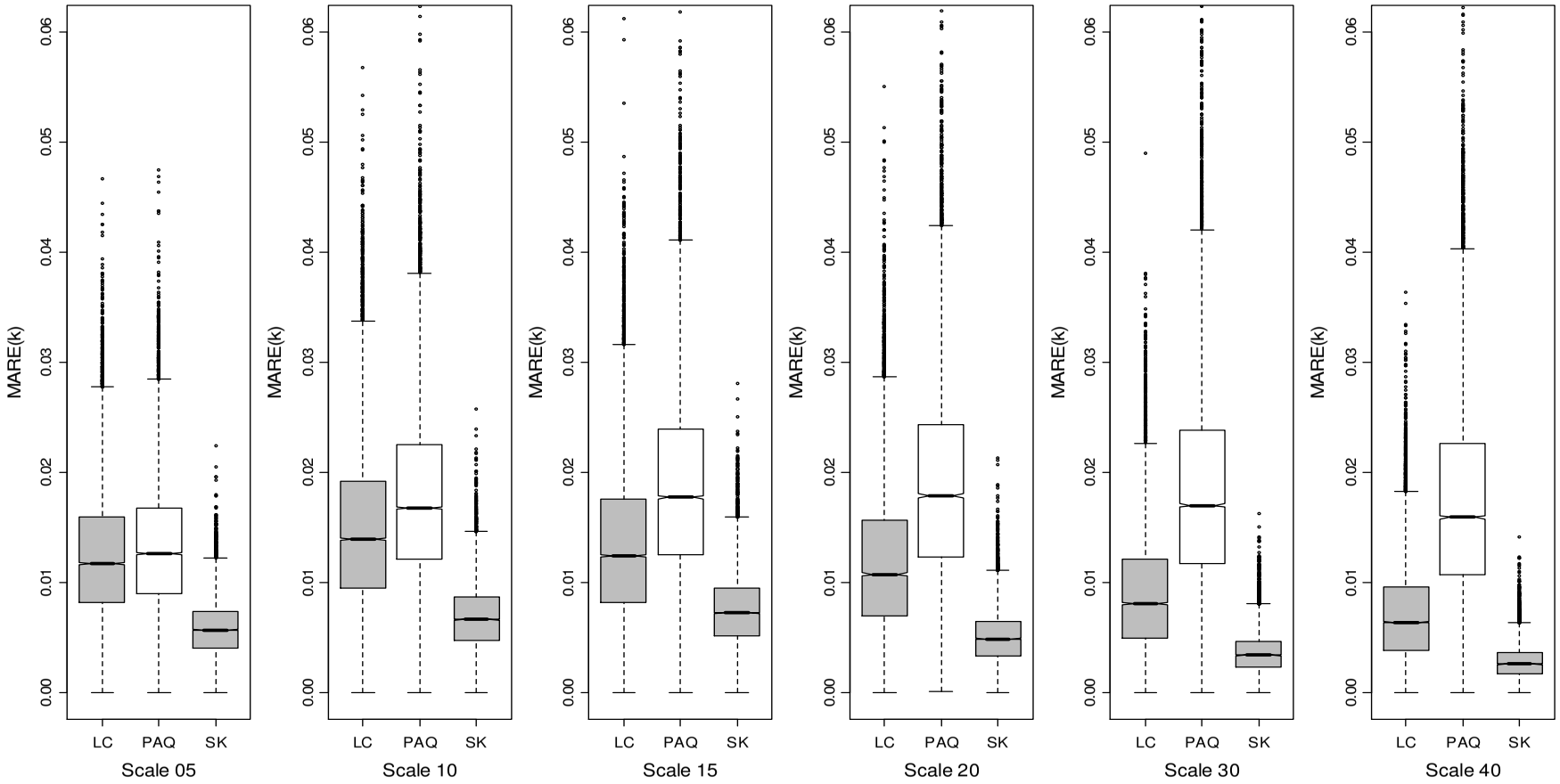


Figure 3.8 - Boxplots for $MARE_k$ values of PAQ, LEACH (LC) and SKATER (SK) according to the scale parameter.

We have made a simple evaluation of the energy cost involved in clusters formation by analyzing the number of clusters assembled, the size of these clusters and the number of solitary nodes. Having few but large clusters is important to save energy, since it minimizes the number of the nodes with the most costly tasks: the heads and the solitary nodes, in this order.

Figure 3.9 presents the summaries for the number of clusters, the size of the clusters and the number of solitary nodes, respectively, for each clustering proposal according to the spatial scale.

Whereas PAQ built few clusters (13, in median, Figure 3.9A) but left many nodes alone (19 to 20, in median, Figure 3.9C), SNAP clustered almost all nodes (41 to 42 clusters, in median, Figure 3.9A) at the cost of having set many heads. DARC and DA-DCA algorithms have built 26 and 22 to 25 clusters (in median, Figure 3.9A), respectively, which we consider to be a trade-off between the number of clusters and their size.

SNAP algorithm produced the smallest number of solitary nodes, followed by DA-DCA algorithm (0 and 1, respectively, Figure 3.9C). SNAP has also produced the smallest clusters (with 2 nodes, in median, Figure 3.9B). On the other extreme, PAQ has left the largest number of nodes without a cluster (19 to 20, in median) and built the largest clusters (6 nodes, in median). In the middle, DARC and DA-DCA built clusters with 3 and 4 nodes, respectively. Considering the network configuration and the radio range we have adopted, the maximum number of nodes in a cluster is 13.

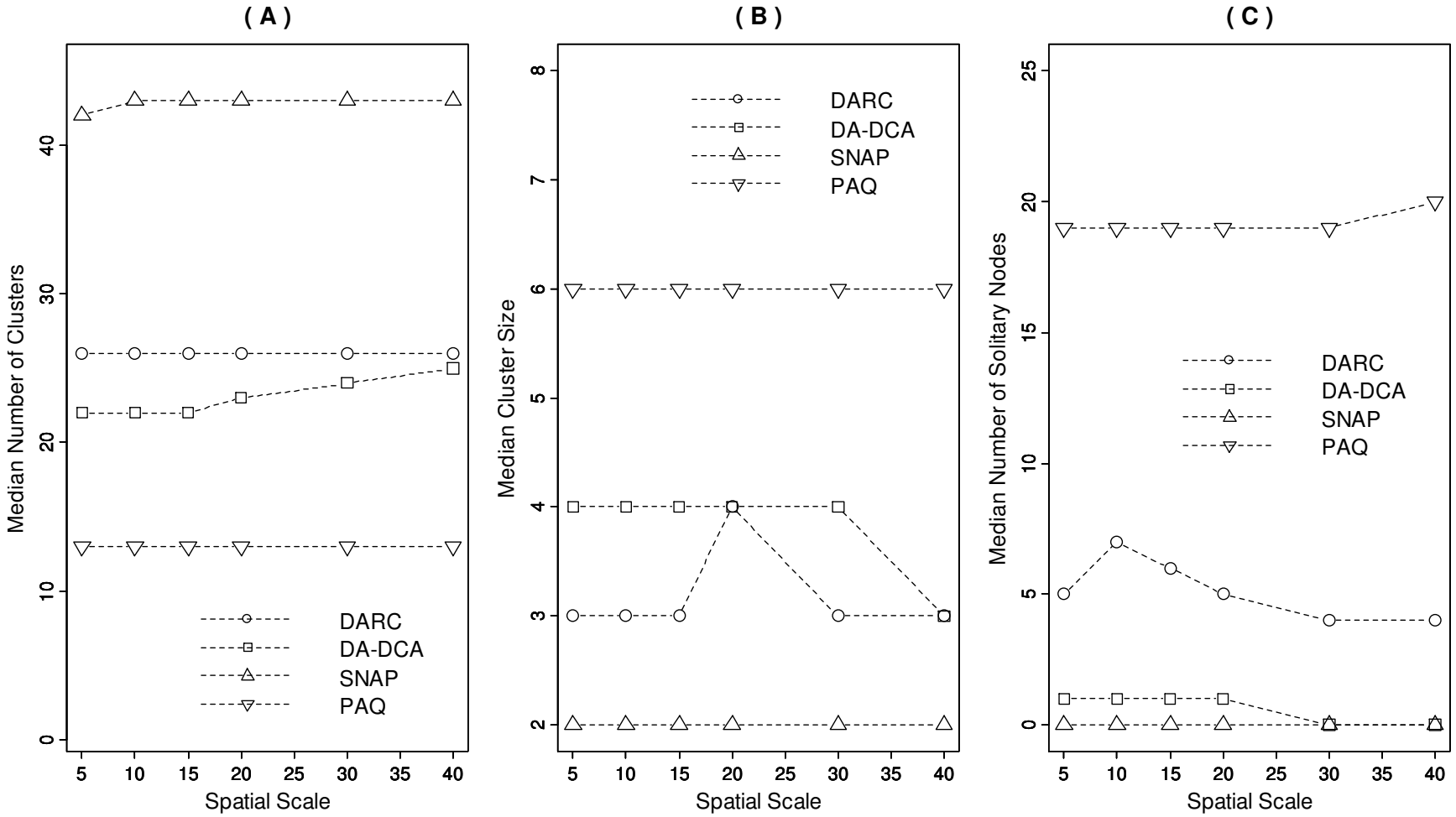


Figure 3.9 - Simplified analysis of the energy costs of the proposals according to the scale parameter (based on 1000 simulations). (A) median number of clusters; (B) median cluster size and (C) median number of solitary nodes.

It is worth to note that the clusters size, the number of clusters and the number of solitary nodes have remained almost constant despite of the increasing in the spatial autocorrelation of the original data. As an exception to this behavior, the number of nodes left alone by DARC algorithm has been sensitive to the increasing of spatial autocorrelation (Figure 3.9C). When the geosensor data have not been able to capture the spatial autocorrelation of the original data (for the smallest spatial scales), the adjusting phase of DARC has not been able to decrease the number of solitary nodes. However, this number has decreased as the spatial autocorrelation increased.

3.4 Final Remarks

In this chapter, we proposed two data-aware clustering algorithms, DARC and DA-DCA. Our goal has been to increase the clusters homogeneity in comparison to the usual proposals. Clusters more homogeneous should produce summaries that would be better estimates for the cluster members' data.

DARC's clusters had the largest internal homogeneity and produced averages with the smallest prediction errors. Furthermore, the evaluation experiments have shown DARC can cluster almost all nodes without overloading an excessive number of them with the head tasks. On the clustering costs, DARC builds the initial clusters spending from three to four local messages. To maintain the clusters, nodes spend from zero to three local messages. In SNAP algorithm (KOTIDIS, 2005), for instance, the nodes must monitor their neighbors' data to estimate the regression models during clusters building and maintenance, besides they spend up to six messages to build the clusters.

Although DA-DCA has produced less costly scenarios than DARC's scenarios (larger number of smaller clusters and less solitary nodes), DARC's clusters have got a better performance (larger internal homogeneity and averages with smaller prediction errors).

A future work includes to associate DARC and DA-DCA algorithm to a scheme for temporal data suppression. This will produce a scheme for spatio-temporal data suppression for geosensor data collection. Therefore, we will be able to evaluate the clusters maintenance.

Although we have designed our proposals to meet the requirements of a spatio-temporal data suppression scheme, it can be adapted for a cluster-based data routing protocol. Moreover, despite of the fact that we have supposed a static network in this chapter, one can adapt DARC and DA-DCA algorithms to the mobile case, since they are procedures that use only the neighbors within the radio range of the nodes.

4 SUPPRESSING TEMPORAL DATA IN SENSOR NETWORKS USING A SCHEME ROBUST TO ABERRANT READINGS♦

4.1 Introduction

Sensor networks are a powerful instrument for data collection, especially for applications like habitat and environmental monitoring. These applications often require continuous updates of the database at the network's root. However, sending continuous reports would quickly run out the limited energy of the nodes. A solution for continuous updating without continuous reporting is to use data suppression (SILBERSTEIN *et al.*, 2007a).

To define a data suppression scheme, nodes and base station have to agree on an expected behavior for the nodes' readings. Thus, nodes only send reports to the base station when their values do not fit to the expected behavior, which is used to predict the suppressed data.

Model-driven data collection (CHU *et al.*, 2006) defines the mean of a node's observations as their expected behavior and models this mean using temporal or spatio-temporal correlations.

A temporal suppression scheme uses the correlation among the readings of a same node to build the expected behavior for the nodes' readings (TULONE and MADDEN, 2006). A spatio-temporal suppression scheme also considers the

♦ This chapter is the manuscript accepted for publication in the *International Journal of Distributed Sensor Networks* (IJDSN). The manuscript is under the final revision.

correlation among the observations of neighboring nodes (SILBERSTEIN *et al.*, 2007a).

Suppression schemes are an alternative to improve the reactivity of a sensor network, which is defined as the ability of a network to react to its environment providing only relevant data (CARDELL-OLIVER *et al.*, 2005). Instead of changing the sampling rates according to the sampled values and sending all collect data to the base station as in CARDELL-OLIVER (2005), a suppression scheme collects data using a constant rate. However, it only sends data if they represent a deviation from the behavior agreed by nodes and base station.

Usually, suppression schemes define an absolute error measure to evaluate the deviation between sensed data and their expected behavior. This produces data collection schemes that are sensitive to aberrant readings. These outlying values can be the result of a temporarily malfunctioning of a particular sensor or due to some intervention on the environment on which the network is operating and it does not have any relation with the monitored variables. Sometimes, aberrant readings can be the result of an expected change in the sensed values. For instance, solar radiation measurements often suffer the effect of temporary clouds. In this case, a reduction in the radiation values is expected and, perhaps, non-interesting to the network user.

Sensors measuring environmental variables can produce such erroneous or nonsense readings (BRANCH *et al.* (2006), KOTIDIS *et al.* (2007), PALPANAS *et al.* (2003) and SUBRAMANIAM *et al.* (2006)), particularly in outdoor applications (SZEWCZYK *et al.* (2004) and TATESON *et al.* (2005)). In monitoring networks with low energy constrains, such as the regular weather stations, the nodes transmit or record the aberrant readings, which are identified and deleted in the base station. However, for a sensor network, transmitting nonsense values means to waste valuable resources.

In this chapter, we propose a temporal suppression scheme that is robust to aberrant readings. Our proposal is based on the detection of outliers and their posterior classification into change-points or aberrant readings. We consider the sequence of data collected by a node as observations of a temporal process. The probabilistic distribution of this process at each time period is used to infer about the expected behavior of the observations. An outlier is an observation that presents a small probability to belong to the distribution at the current time period. An outlier reading may suggest a change in the expected value for the time series or it may be an aberrant reading.

To detect outliers from a time series, we have adapted the proposal in YAMANISHI and TAKEUCHI (2002). We have inserted our version as part of a suppression scheme for data collection in sensor networks, the TS-SOUND scheme (Temporal Suppression by Statistical OUtlier Notice and Detection). After detecting an outlier, TS-SOUND classifies it into a change-point or an aberrant reading. In the former case, the node sends data to the base station. Otherwise, the node suppresses its data.

We have designed TS-SOUND for applications that are not interested in aberrant readings, since they represent a failure in data sensing or processing. Usually, these erroneous measurements occur at random, isolated or clustered. If they remain, this means malfunctioning and suggests a non reliable node.

TS-SOUND scheme adopts a procedure to avoid detecting an aberrant reading as a change-point. Furthermore, even if this misdetection occurs, TS-SOUND does not send the aberrant reading to the base station.

In this chapter, we claim and demonstrate that our proposed scheme for temporal suppression data is robust to aberrant readings. Furthermore, considering the trade-off between energy consumption and data quality, TS-SOUND has

outperformed the model-based suppression schemes we have considered in this chapter (PAQ (TULONE and MADDEN, 2006) and exponential regression (SILBERSTEIN *et al.*, 2007a)) and also the simplest data suppression scheme, VB scheme (SILBERSTEIN *et al.*, 2007a). The prediction error measures the quality of the data sent to the base station. Since the data transmission is the most important energy consumer, we use the suppression rates as a proxy for the energy consumption. To evaluate TS-SOUND scheme, we have run evaluation experiments with real and simulated data. The real data have come from several sources and presented different behaviors.

The remainder of this chapter is organized as follows. Section 4.2 presents a TS-SOUND overview. In section 4.3, we describe the related work and the framework for suppression schemes proposed in SILBERSTEIN *et al.* (2007a). Section 4.4 describes SDAR algorithm (YAMANISHI and TAKEUCHI, 2002), which allows for the on-line estimation of time series parameters. In addition, it describes the procedure in (YAMANISHI and TAKEUCHI, 2002) to detect outliers, how we have adapted it to be part of our proposed suppression scheme and how TS-SOUND deals with classifying the outliers into change-points or aberrant readings. In section 4.5, we present TS-SOUND protocol and frame it as a suppression scheme according to the proposal in SILBERSTEIN *et al.* (2007a). Section 4.6 describes the evaluation experiments and section 4.7 presents their results using real and simulated data. Finally, section 4.8 discusses the experiments results and section 4.9 presents some future directions.

4.2 TS-SOUND overview

Techniques for outlier detection have been proposed in communities such as Statistical Process Control (for example, FRISÉN (2003) and POLLAK and SIEGMUND (1991)), Data Mining, Database and Machine Learning (for example, HODGE and AUSTIN (2004) ; MUTHUKRISHNAN *et al.* (2004) ; RAMASWAMY *et al.* (2000) ; YAMANISHI and TAKEUCHI (2002)).

In Statistical Process Control (SPC), the goal is to monitor a process initially "in-control" and raise an alarm when this process is considered to be "out-of-control" as soon as possible. Often, the "in-control" state of the process is a predefined condition: nominal values for the monitored parameters and their tolerance bounds. To raise the alarm, SPC uses procedures to detect outliers.

For TS-SOUND, the "in-control" state is the probabilistic distribution of the monitored variable at the last time period. If the process is "in-control" during a time interval, the sensor readings follow the same probabilistic distribution along this interval and different values are caused by random fluctuation around an expected value. Then, we can suppress these readings. We consider the process is "out-of-control" if the expected value of this distribution changes. After the change, a new "in-control" state is defined. The change's relevancy is a user-defined parameter.

As in the SPC techniques, TS-SOUND uses the outlier occurrence to infer if the process is "out-of-control". To detect outliers, TS-SOUND adapts the technique in YAMANISHI and TAKEUCHI (2002), which has been proposed to detect outliers from a time series. TS-SOUND employs an algorithm that considers the temporal dependence of the time series to update the parameters of the probability distribution at each new sensor reading (on-line estimation). This algorithm is called SDAR (Sequentially Discounting Auto-Regressive) (YAMANISHI and TAKEUCHI, 2002). SDAR combines the last parameters' updates with the new sensor reading to produce the new parameters' updates. SDAR uses a discounting factor to control the weight of the new sensor data in the updates' values. The outliers are detected as deviations from the data distribution.

In a time series, an outlier can suggest a distribution change-point or an aberrant reading. We can distinguish a change-point from an aberrant reading if we compare the time series values before and after the outlier, examining, for

instance, the time series plot (Figure 4.1). The aberrant points appear as the "peaks" or "spikes" of the time series plot. The time series has similar behaviors before and after the occurrence of aberrant readings. On the other hand, after a change-point, the time series changes its behavior. Then, a data suppression scheme must update the database at the base station only when change-points occur.

To distinguish change-points from aberrant readings, TS-SOUND opens a post-monitoring window whenever it detects an outlier. During this time interval, the node goes on collecting data and updating the estimated parameters. At the end of this time window, TS-SOUND compares the collected values with the distribution before and after the detected outlier. This outlier is classified as a change-point if the post-monitoring data are considered to be: 1) discrepant readings in relation to the distribution before the outlier; 2) non discrepant readings in relation to distribution after the outlier. If TS-SOUND classifies the detected outlier as a change-point, it summarizes the data collected during the post-monitoring and sends the result to the base station.

We have adopted a post-monitoring window for two reasons: a) to be able to distinguish change-points from aberrant readings. It avoids sending the latter ones to the base station; b) to allow for capturing the value of the new expected behavior through the summary of the collected values.

The base station uses the last sent data as an estimate for the node's readings until it receives a message with new data. Thus, for each node in the network, the base station stores a sequence of summaries and uses this time series as an estimate for the real node's time series. Section 4.5 describes TS-SOUND suppression scheme in detail.

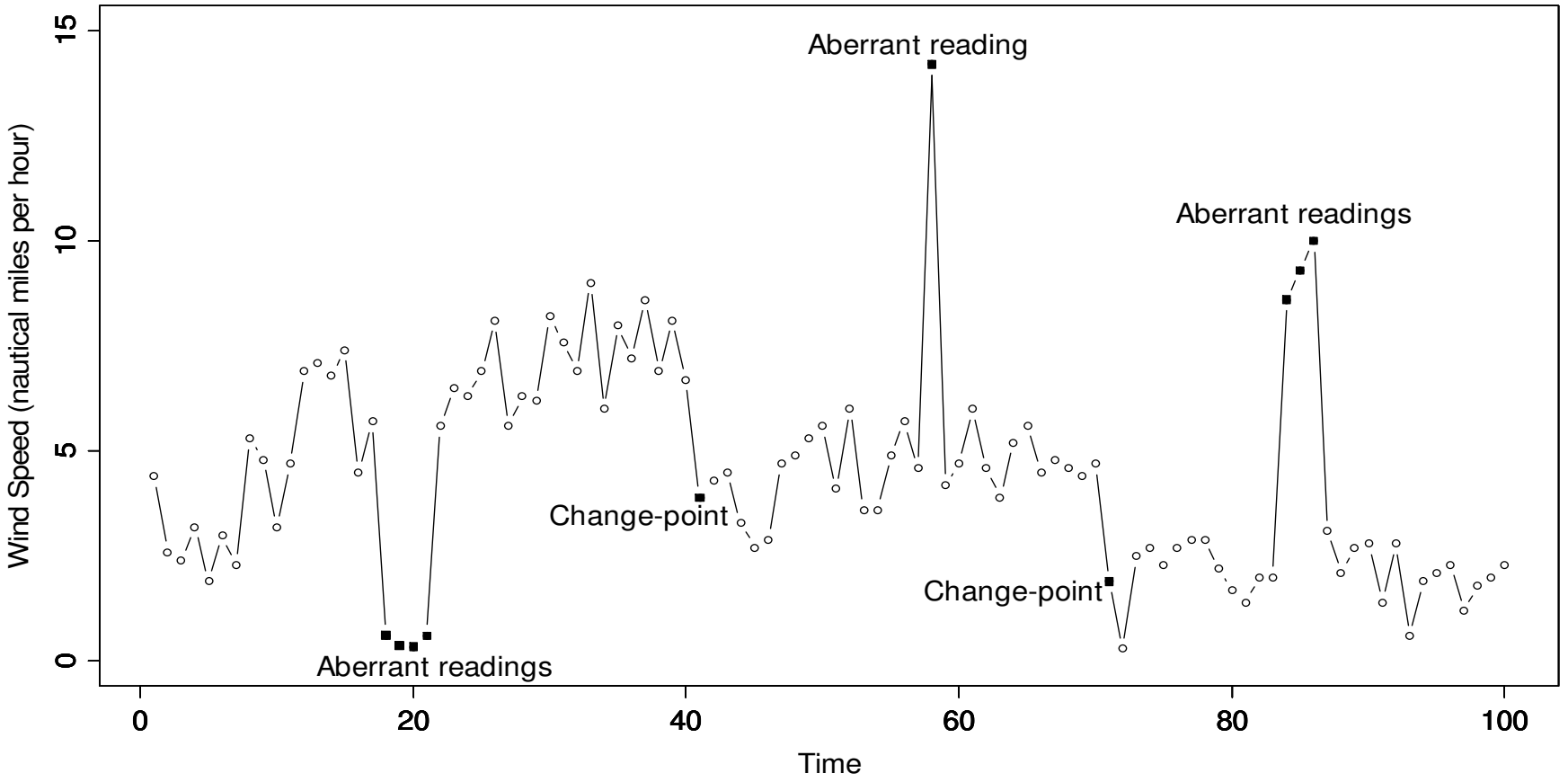


Figure 4.1 - Outliers in a wind speed time series (black dots).

Source: weather station of the University of Washington, USA, October 2006. We have inserted the aberrant readings to produce this figure.

4.3 Related Work

Recently, some protocols for data suppression in sensor networks have proposed to use statistical models to predict the nodes' data at the base station reducing the amount of communication inside the network. This approach is called model-driven data suppression (CHU *et al.*, 2006).

The main idea in CHU *et al.* (2006) is to keep synchronized two probabilistic models: one at base station and other at the nodes. The model parameters are estimated in a learning phase. Based on these identical models, nodes and base station make the same predictions on the data to be collected. Then, the node collects the actual data and compares them to its prediction. If the difference between the real and predicted values is greater than a user-defined error bound, the node sends its data to the base station. Otherwise, the node suppresses the collected data.

A similar idea appears in TULONE and MADDEN (2006). PAQ protocol makes predictions based on a time series model, the third-order autoregressive model, AR(3). Given a time period t, the predicted value in t is written as a linear combination of the last three observations before t. PAQ uses two predefined error bounds to monitor the prediction error, which is defined as the absolute difference between the real and the predicted value. When the prediction error is greater than ε_υ, PAQ considers the observation as an outlier and sends it to the base station. If the prediction error is smaller than ε_υ but it is greater than ε_δ ($\varepsilon_\delta < \varepsilon_\upsilon$), PAQ opens a monitoring window. During the next A_{PAQ} time periods, the node goes on collecting data, predicting their values, monitoring outliers and sending them to the base station. At the end of the monitoring window, PAQ counts how many observations have had prediction errors greater than ε_υ or greater than ε_δ but smaller than ε_υ. If this sum is greater than a threshold a ($a \leq A_{PAQ}$), PAQ decides to relearn the four model parameters. Then, PAQ calculates their new values and sends them to the base station. A variation of PAQ, called in

SILBERSTEIN *et al* (2007a) as exponential regression (EXP), uses the observation in the time period (t -1) in a simple linear regression to predict the observation in t. Thus, EXP has to estimate two model parameters.

It is worth to mention that neither PAQ nor EXP distinguish a change-point from an aberrant reading. Once they detect an outlier reading, the node sends the observation to the base station, even if it is an aberration.

4.3.1 Temporal Suppression Schemes

Silberstein et al. (2007a) defined a general framework for suppression schemes. This framework makes easier the comparisons among data suppression schemes.

The nodes in the network are classified into "updaters" and "observers". A suppression link describes the suppression/reporting relationship between an updater and its observer. The set of suppression links within the sensor network defines a suppression scheme.

In a simple suppression scheme, all the network nodes are updaters. These updaters collect data and decide to send them (or not) to the observer node, which is the base station. To produce a report r_t to its observer, the updater uses an encoding function f_{enc}. To decode the updater report, the observer uses a decoding function.

The vector $\mathbf{X}_t$ represents the data of the updater node at time period $\underline{t}$ and the vector $\hat{\mathbf{X}}_t$ represents the data as calculated by the observer node at same time

period[18]. The suppression link maintains $\mathbf{X}_t$ and $\hat{\mathbf{X}}_t$ synchronized by evaluating a function $g(\mathbf{X}_t;\hat{\mathbf{X}}_t)$. The function g returns the loginal true value if $\hat{\mathbf{X}}_t$ is within a user-defined error tolerance (**ε**) of $\mathbf{X}_t$.

In Value-Based (VB) suppression scheme, for instance, the encoding and decoding functions are defined, respectively, by

$$f_{\text{enc}} = \begin{cases} x_t - x_{t'}, & \text{if } |x_t - x_{t'}| > \varepsilon_{VB} \\ \perp, & \text{otherwise} \end{cases} \quad \text{and} \tag{4.1}$$

$$\hat{x}_t = \begin{cases} \hat{x}_{(t-1)} + r_t & \text{if } r_t = x_t - x_{t'} \\ \hat{x}_{(t-1)}, & \text{if } r_t = \perp \end{cases}, \tag{4.2}$$

where x_t is a component of the vector $\mathbf{X}_t$, *t'* is the last time the updater sends a message to its observer and the symbol $\perp$ represents data suppression. The value $x_{t'}$ is what the observer knows about its updater at time period *t*. If the relative difference between the current updater value x_t and $x_{t'}$, the g function, is greater than error bound ε_{VB}, the updater produces a report $r_t = x_t - x_{t'}$ and sends it to the observer node. Otherwise, no message is sent ($r_t = \perp$). The observer computes its value $\hat{x}_t$ by adding the received report r_t to its old value $\hat{x}_{t-1}$. If the updater does not send a message, the observer updates $\hat{x}_t$ by repeating the old value.

PAQ and exponential regression have also been framed as temporal suppression schemes. Although PAQ also has a proposal for spatio-temporal suppression

[18] The authors use a vector to represent the data of a node because this node can have more than one value to send to the base station. That is the case of PAQ suppression scheme, for example.

(TULONE and MADDEN, 2006), we just consider its temporal version in this chapter. The expressions in (4.3) and (4.4) reproduce the encoding functions of PAQ and EXP, respectively,

$$f_{\text{enc}} = \begin{cases} \alpha_t, \beta_t, \gamma_t, \eta_t & \text{if } (\texttt{modelRelearn}) \\ x_t & \text{if } (\texttt{outlier}) \\ \perp & \text{otherwise} \end{cases} . \quad (4.3)$$

$$f_{\text{enc}} = \begin{cases} \alpha_t, \beta_t & \text{if } (\texttt{modelRelearn}) \\ x_t & \text{if } (\texttt{outlier}) \\ \perp & \text{otherwise} \end{cases} . \quad (4.4)$$

In (4.3), α_t , β_t , γ_t and η_t are the coefficients of the AR(3) model adopted by PAQ scheme and, in (4.4), α_t and β_t are the coefficients of the simple linear regression model adopted by EXP scheme. The functions `modelRelearn` and `outlier` enclose the g function of PAQ and EXP schemes. As in VB scheme, it also evaluates the error between real and predicted values.

We classify our TS-SOUND proposal as a model-driven approach for temporal suppression (SILBERSTEIN *et al.*, 2007a). TS-SOUND models the mean of the monitored variable and uses it to decide if an observation is an outlier of the current data distribution. However, the model runs only at the nodes, not at the base station, being not necessary to keep synchronized models as in the other model-driven proposals. We frame TS-SOUND approach as a temporal suppression scheme in section 4.5.

4.3.2 Outliers detection in a sensor network

The problem of detecting outliers in a sensor network has gained importance in proposals such as in BRANCH (2006), KOTIDIS *et al.* (2007), PALPANAS *et al.* (2003) and SUBRAMANIAM *et al.* (2006). The proposal in KOTIDIS *et al.* (2007) removes outlier readings from the data aggregation and makes them available to the monitoring application. In SUBRAMANIAM *et al.* (2006), the authors detect

outliers within a sliding window that holds the last W values of the sensor data. To estimate the data distribution, they use nonparametric models. Moreover, they report the outlier readings in a hierarchical structure, using the union of the outliers coming from multiple sensors. Branch et al. (2006) propose a generic distributed algorithm that accommodates many nonparametric methods to detect outliers such as "distance to the k-th nearest neighbor" and "average distance to the k nearest neighbors". Nodes use one of these techniques to find out their local outliers. Then, they exchange information about these local outliers with their neighboring nodes to find out global outliers. Palpanas et al. (2003) use kernel density estimators to approximate the data distribution at each sensor node. As SDAR algorithm in YAMANISHI and TAKEUCHI (2002), the kernel density estimation allows for adjusting itself to the input data distribution, as this distribution changes overtime. The proposal in PALPANAS *et al.* (2003) assumes a heterogeneous sensor network, in which few sensor nodes are more powerful than the other sensors in the network. The detection of outliers is performed by these empowered nodes, which combine the models of two or more sensor nodes in this task. The authors discuss the trade-off among data accuracy, number of updates and the size of estimation models in some application scenarios. However, they do not provide evaluation experiments to show how this would work on real data.

Differently from the proposals described above, our proposal to detect outliers does not require communication among sensor nodes, since we have treated only the temporal aspect of the data suppression in this chapter. However, some of these proposals can be an interesting basis for a future spatio-temporal version of TS-SOUND scheme.

4.4 Detecting outliers from a time series

In this section, we present the procedure in YAMANISHI and TAKEUCHI (2002) to detect outliers from a time series and our proposal for adapting it to the constrained environment of a sensor network.

We consider the sequence of the data sensed by a sensor node, $\{X_t, t=1,2,3\ldots\}$, as a time series.

The autoregressive (AR) model is the simplest model to represent the statistical behavior of a time series. In AR(k), the autoregressive model of order k, the observation at time t, X_t, is written as a combination of the last k past observations as following

$$X_t = \mu + \rho_1(X_{t-1} - \mu) + \rho_2(X_{t-2} - \mu) + \ldots + \rho_k(X_{t-k} - \mu) + \varepsilon_t, \quad k=1,2,3,\ldots,t\text{-}1 \qquad (4.5)$$

where μ is the mean of X_t, ρ_k is the autocorrelation of order k and ε_t is a noise term following a Gaussian distribution with zero mean and variance σ_ε^2 .

To simplify the calculations in the sensor nodes, we have adopted the AR(1) model. From now on, we use this model to present the approach in YAMANISHI and TAKEUCHI (2002).

If we use an AR(1) model to represent the time series, the probability density function of X_t, given X_{t-1}, is

$$p_t\left(X_t \mid X_{t-1}; \theta^t\right) = \frac{1}{\sigma^t\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{X_t - w^t}{\sigma^t}\right)^2\right], \qquad (4.6)$$

where $w^t = \mu^t + \rho_1^t\,(X_{t-1} - \mu^t)$ is the prediction for X_t using the AR(1) model, $\rho_1^t = C_1^t / C_0^t$ is the autocorrelation between X_t and X_{t-1}, $\left(\sigma^t\right)^2 = C_0^t - \rho_1^t C_1^t$, C_0^t is the

variance of X_t, C_1^t is the covariance between X_t and X_{t-1} and $\theta^t = (\mu^t, \rho_1^t, \sigma^t)$ is the parameters vector. In other words, $[X_t | X_{t-1}]$ follows the Gaussian distribution with mean w^t and variance $(\sigma^t)^2$.

4.4.1 The Yamanishi and Takeuchi's proposal to detect outliers

Yamanishi and Takeuchi (2002) adopted the AR model to represent the time series.

To estimate the parameters in θ and, as a result, the value for $p_t(X_t \mid X_{t-1}; \theta^t)$, Yamanishi and Takeuchi (2002) proposed the *Sequentially Discounting AR* (SDAR) algorithm. The goal of SDAR is to learn of the AR model and provide the *on-line* estimation of θ, which is updated at each new observation X_t. A *discounting factor* r controls the weight given to the new observation X_t in the estimation of θ.

SDAR has two main steps: initialization and parameters updating. In the first step, SDAR sets $\mu^0, C_0^0, C_1^0 \rho_1^0$ and σ^0, which are the initial values for $\mu^t, C_0^t, C_1^t \rho_1^t$ and σ^t, respectively. The initial values for μ^t, C_0^t and C_1^t can be defined by the user or calculated using a learning sample.

The second step of SDAR is parameters updating. At each time ***t***, the node collects a new observation X_t and, for a given value of r, $0 \leq r \leq 1$, the parameters are updated as following:

$$\hat{\mu}^t = (1-r)\hat{\mu}^{t-1} + r X_t \text{ ,} \tag{4.7}$$

$$\hat{C}_j^t = (1-r)\hat{C}_j^{t-1} + r(X_t - \hat{\mu}^t)(X_{t-j} - \hat{\mu}^{t-j}) \text{ , } j = 0,1. \text{ ,} \tag{4.8}$$

$$\hat{\rho}_1^t = \hat{C}_1^t \Big/ \hat{C}_0^t \text{ ,} \tag{4.9}$$

$$\hat{w}^t = \hat{\rho}_1^t (X_{t-1} - \hat{\mu}^t) + \hat{\mu}^t \quad \text{,} \tag{4.10}$$

$$\left(\hat{\sigma}^{t}\right)^{2} = (1-r)\left(\hat{\sigma}^{t-1}\right)^{2} + r\left(X_{t} - \hat{w}^{t}\right)^{2}. \qquad (4.11)$$

The discounting factor r enables SDAR to deal with nonstationary time series.

Since SDAR updates the parameters at each time t, it produces a sequence of probability densities $\{ p_t$, t=1,2,3…$\}$, where p_t is the probability density function in (4.6) specified by the parameters updated by the SDAR algorithm at time t.

To detect outliers, the authors have proposed to evaluate each observation X_t using the sequence $\{ p_t$, t=1,2,3…$\}$ and the score function

$$score(X_t) = -\ln\left[p_{t-1}(X_t)\right] = \frac{1}{2}\left(\frac{X_t - w^{t-1}}{\sigma^{t-1}}\right)^{2} - \ln\left[\frac{1}{\sigma^{t-1}\sqrt{2\pi}}\right] \qquad (4.12)$$

Intuitively, this score measures how large the probability density function p_t has moved from p_{t-1} after learning from X_t. A high value for $score(X_t)$ indicates X_t is an outlier with a high probability.

To detect change-points, Yamanishi and Takeuchi (2002) proposed to use the average of the T last values of $score(X_t)$, $T > 1$, to construct a time series Y_t. Then, SDAR algorithm is applied on Y_t to construct a sequence of probability densities q_t and $score(Y_t) = -\ln\left[q_{t-1}(Y_t)\right]$ is calculated. Then, they define a function $Score(t)$, which is the average of the T′ last values of $score(Y_t)$, T′$>$1, and use $score(Y_t)$ to detect change-points in the time series.

It is worth to note there are many calculations involved in the proposal in YAMANISHI and TAKEUCHI (2002). Moreover, they have not made clear how to distinguish aberrant readings from change-points.

4.4.2 The outlier detection in the TS-SOUND scheme

TS-SOUND scheme uses the detection of outliers to decide whether a node must suppress its data or it must not. If an outlier is detected, the node opens a post-monitoring window to decide if the outlier is a change-point or an aberrant reading. In the first case, the node sends data to the base station.

Yamanishi and Takeuchi (2002) have not considered power limitations in the calculations. Therefore, using a logarithm operator in $score(X_t)$ has not been a concern. However, in the constrained environment of a sensor node, using the logarithm function can be a costly operation. Then, to meet the requirements of a scheme for data collection in sensor networks, we have simplified the definition of $score(X_t)$ by evaluating the distance between X_t and $\hat{w}^{t-1}$ using the function

$$SD_{t-1}(X_t) = \frac{\left| X_t - \hat{w}^{t-1} \right|}{\hat{\sigma}^{t-1}}, \tag{4.13}$$

where $\hat{\sigma}^t$ represents the estimate for the standard deviation of X_t.

Note that $SD_{t-1}(X_t)$ is the absolute value of a normalized score. In fact, we can see $SD_{t-1}(X_t)$ as part of G statistic[19] proposed in (GRUBBS, 1969) to detect outliers in a static dataset. As the original $score(X_t)$ in (4.12), $SD_{t-1}(X_t)$ evaluates how far X_t is from $\hat{w}^{t-1}$, which is the prediction for X_t using the AR(1) model in *t-1*. Then, a high value for $SD_{t-1}(X_t)$ also indicates X_t is an outlier of the distribution in *t-1* with a high probability.

[19] G statistics is defined as the maximum of the absolute value of the normalized scores of observations in a static dataset.

As in YAMANISHI and TAKEUCHI (2002), we evaluate the $SD_{t-1}(X_t)$ function over a time window composed by the T past time periods, where $T \geq 1$. However, instead of using a T-averaged score, we simplify the calculations and use the sum of the T past values of $SD_{t-1}(X_t)$. Then, at each time period t, we calculate the score Z_t as

$$Z_t = \sum_{i=t-T+1}^{t} SD_{i-1}(X_i) = \sum_{i=t-T+1}^{t} \frac{\left| X_i - \hat{w}^{i-1} \right|}{\hat{\sigma}^{i-1}} \qquad (4.14)$$

The expression for Z_t compares the values of $\{X_i, i=t-T+1, \ldots, t\}$ with $\hat{w}^{t-1}$, which is the predicted value for them if they come from the p distribution in $t=\underline{i}-1$. Large differences indicate the values of $\{X_i, i=t-T+1, \ldots, t\}$ have a small probability to belong to the p distribution in $t=i-1$. The sum over the T past time periods in Z_t allows for capturing smooth changes in the average of the time series.

If the value of Z_t is greater than a pre-defined threshold, X_t is considered to be an outlier. However, X_t can be an aberrant reading or a change-point. To decide this, TS-SOUND scheme opens a post-monitoring window.

4.4.2.1 The threshold for Z_t

Besides simplifying the calculations of Z_t, the scoring function $SD_{t-1}(X_t)$ makes the definition of a threshold for Z_t more intuitive than choosing a threshold to the original $Score(X_t)$ in YAMANISHI and TAKEUCHI (2002). We have used the theory of statistical significance tests (LEHMAN, 1997) to help us with this choice.

At each time period t, we can see the classification of X_t as an outlier of the p distribution in t-1 as a significance test of the following hypothesis

H_0: the expected value for X_t is w^{t-1} (X_t is not an outlier) versus
H_1: the expected value for X_t is not w^{t-1} (X_t is an outlier).

At a significance level of α, $0 < \alpha < 1$, the null hypothesis H_0 is rejected if $\left|Z_{test}^{t}\right| > z_{\alpha/2}$, where $Z_{test}^{t} = \frac{X_t - \hat{w}^{t-1}}{\hat{\sigma}^{t-1}}$ is a normalized score and $z_{\alpha/2}$ is the percentile 100(1-α/2) of the standard Gaussian distribution (average and standard deviation equal to 0 and 1, respectively). Here, we assume the estimates for w^{t-1} and σ^{t-1} carry enough information from the past data to approximate the distribution of Z_{test} by a standard Gaussian distribution.

Since Z_t is the sum of $\left|Z_{test}^{i}\right|$, $i=t-T+1, \ldots, t$, one can use the Gaussian model with average equals to zero and standard deviation equals to $\sqrt{T}$ to *guide* the choice of the values for z_T^{α}, the threshold for Z_t. For instance, if $T = 2$ and the significance levels α = (0.20, 0.10, 0.05, 0.025, 0.01), the values for z_T^{α} would be 1.81, 2.32, 2.77, 3.17 and 3.64, respectively. These are the values of the percentiles 100(1-α/2) of a Gaussian distribution with mean and standard deviation equal to 0 and $\sqrt{2}$, respectively.

It is worth to note that the terms $\left|Z_{test}^{i}\right|$, i=t-T+1, …, t, are not independent. Assuming they are positively correlated, $Var\left(\sum_{i=t-T+1}^{t}\left|Z_{test}^{i}\right|\right) < T$. Then, the values of z_T^{α} should be smaller than they will be if we assume the independence and use $Var\left(Z_t\right) = T$. This makes harder the detection of X_t as an outlier. However, accouting for the dependence in this case is not a trivial task. Then, we expect the choice for the values of α can help to minimize this problem.

The value of z_T^{α} depends on two user-defined parameters: the size of the risk of making a mistake when the scheme classifies X_t as an outlier (α) and how much of the past observations should be considered in this classification *(T)*. For a fixed value of *T*, the smaller the value of α, the more rigorous the criterion to consider X_t as an outlier of the

distribution in *t-1*. Then, decreasing the value of α increases the value of z_T^{α} and, as a result, the data suppression rate increases.

For a fixed value of α, the greater the value of *T* is, the greater the delay to detect an outlier. On the other hand, increasing the value of *T* allows for capturing smooth changes in the expected value for the time series. The relevance of the change is a user-defined parameter and also has to do with the value for α: if α is large, the scheme will be able to detect small changes, since the outlier alarm will rise more often.

In our experiments, we have evaluated the values α = (0.25, 0.20, 0.15, 0.10, 0.05, 0.025, 0.01) and T = (2, 4, 6, 8, 10). We discuss these values using a simple case study in section 4.7.1.

4.4.2.2 Detecting change-points

After detecting an outlier at time period *t*, TS-SOUND has to classify it as a change-point or an aberrant reading. To make this decision, the node has to study the time series behavior before and after *t*. Then, if TS-SOUND detects an outlier, it opens a post-monitoring window of size *T*. From $t+1$ to $t+T$, the node collects data and updates the AR(1) parameters. At the end of post-monitoring window, the node compares the T observations collected during the time window with the *p* distribution *before* and *after* the detected outlier.

As we discussed at section 4.2, the outlier detected at time period *t* is considered to be a change-point if the observations within the monitoring window are considered to be outliers of the *p* distribution *before t* and non-outliers of the *p* distribution *after t*. In Figure 4.1, we can visualize the reason for this rule.

To make the "before-comparison", we use the function Z_{t+T}^{B} defined as following

$$Z_{t+T}^{B} = \sum_{i=t+1}^{t+T} \frac{\left| X_i - w^{(i-1)-T} \right|}{\hat{\sigma}^{(i-1)-T}}. \quad (4.15)$$

Note that Z_{t+T}^{B} uses the estimates for the AR(1) parameters of time periods from t - T to t -1, that is, the last T estimates *before* the detected outlier.

The "after-comparison" is made using the function Z_{t+T}^{A} defined as

$$Z_{t+T}^{A} = \sum_{i=t+1}^{t+T} \frac{\left| X_i - w^{t} \right|}{\hat{\sigma}^{t}} \ . \quad (4.16)$$

The expression for Z_{t+T}^{A} uses the estimates for the AR(1) parameters calculated when the outlier was detected, at time period t.

Then, X_t is considered to be a change-point if $Z_{t+T}^{B} \geq Z_{T}^{c \cdot \alpha}$ and $Z_{t+T}^{A} \leq Z_{T}^{c \cdot \alpha}$, where $0 < c \leq 1$. If $c < 1$, the rigor to consider the observations after t as outliers is greater than the rigor used to detected the outlier in t. Actually, we propose to keep the same rigor level for the "before-comparison" (c=1) and increase the rigor for the "after-comparison" (e.g., c=0.05). This strategy takes into account the values produced immediately after a change-point are possibly accommodating themselves around the new expected value. This can produce values for Z_{t+T}^{A} larger than they should be if a longer time period had been observed, which would lead to the wrong classification of a change-point as an aberrant reading. Then, increasing the rigor in the "after-comparison" decreases the probability of making this mistake.

If the detected outlier is considered to be a change-point, the node updates the database at the base station sending a summary of the observations collected during the post-monitoring window. We have adopted the median to calculate this summary,

since the median is more robust to aberrant readings than the average, for instance. This property of the median can be especially useful if TS-SOUND mistakes the beginning of sequence of aberrant readings for a change-point. In this case, the node will send the summary to the base station unnecessarily, which will degrade the suppression rate. However, the median will suffer less influence of these erroneous readings, especially if the length of the monitoring window is larger than the size of the aberrant sequence. Then, at least the prediction error at base station will be preserved.

It is worth to mention that the length of the post-monitoring window *(T)* could be different from the number of past observations used in SDAR parameters estimation and in Z_t statistics. However, in our additional experiments to evaluate this possibility, TS-SOUND has got the best results when both time windows have had the same length.

4.4.3 Other proposals to detect outliers in a time series

There are other proposals for outliers detection in time series such as GRUBBS (1969), POLLAK and SIEGMUND (1991), RAMASWAMY *et al.* (2000), MUTHUKRISHNAN *et al.* (2004), SUBRAMANIAM (2006) and those described by Hodge and Austin (2004). However, we have considered the proposal in YAMANISHI and TAKEUCHI (2002) as the best one to be adapted to a scheme of data suppression in sensor networks. The reasons for this choice have been the following: a) the proposal in YAMANISHI and TAKEUCHI (2002) considers the temporal autocorrelation of sensor data by adopting a time series model; b) it is adaptative to nonstationary data sources; c) it allows for on-line detection of outliers and d) the calculations can be made simpler.

4.5 TS-SOUND scheme

The TS-SOUND scheme has two phases: learning and operation. In the learning phase, TS-SOUND estimates the initial values for the SDAR parameters and the first two values

for Z_t.

4.5.1 Learning phase

Before beginning its operation, the node collects values during a short time window, say, N_{ini} time periods. The values for the initial values μ^0, C_0^0, C_1^0 are calculated as following

$$\mu^0 = \frac{\sum_{t=1}^{N_{ini}} X_t}{N_{ini}}, \quad C_0^0 = \frac{\sum_{t=1}^{N_{ini}} \left(X_t - \mu^0\right)^2}{N_{ini} - 1}, \quad C_1^0 = \frac{\sum_{t=2}^{N_{ini}} \left(X_t - \mu^0\right)\left(X_{t-1} - \mu^0\right)}{N_{ini} - 1} . \qquad (4.17)$$

To calculate the first value for Z_t, the node needs T additional observations. Then, the size of learning sample is $N_{learn} = N_{ini} + T$. Figure 4.2 presents the pseudo-code for the algorithm running in the learning phase.

Until completing N_{ini} observations, the node collects and stores data every t_s time units, which is the user set sampling rate (lines 1-5).

Discrepant values can affect the estimative for the initial values. Then, the learning algorithm filters these outliers before calculating the initial values. The outliers limits (OUT_{UPPER} and OUT_{LOWER}) are calculated according to the rules for building boxplots (TUKEY, 1977). First, we calculate P_{25} and P_{75}, which are the 25th and the 75th percentiles of the observations, respectively. To calculate the percentiles, the algorithm has to sort the data, which can be done during the values storage. The difference $IQ=(P_{75}-P_{25})$ is called *interquartile range*. The upper and lower limits are defined as $OUT_{UPPER} = (P_{75} + 1.5\ IQ)$ and $OUT_{LOWER} = (P_{25} - 1.5\ IQ)$. Values outside these limits are considered to be outliers.

After removing the possible outliers (lines 7-8), the algorithm calculates the initial values for SDAR parameters (line 9).

```
learning()

Input      r, T, N_ini
Output     initial values for SDAR parameters : μ̂^j, Ĉ_0^j, Ĉ_1^j, ρ̂_1^j, σ̂^2 j  and Z_t.
1)  j=1
2)  every t_s time units while j ≤ N_ini do
3)        read x_j;
4)        enqueue X = X ∪ x_j ;
5)        j= j +1 .
6)  calculate OUT_UPPER , OUT_LOWER .
7)  from j=1 to j=N_ini do
8)    if (OUT_LOWER < X_j < OUT_UPPER) enqueue X_noOut = X_noOut ∪ X_j .
9)  calculate μ^0, C_0^0, C_1^0, ρ_1^0, and σ^2 0 using X_noOut .
10) j = N_ini + 1
11) read x_j;
12) enqueue X = X ∪ x_j ;
13) send x_j;
14) calculate the SDAR parameters μ̂^j, Ĉ_0^j, Ĉ_1^j, ρ̂_1^j, and σ̂^2 j;
15) j = j+1;
16) every t_s time units while j ≤ N_ini + T do
17)   read x_j;
18)   enqueue X = X ∪ x_j ;
19)   calculate and store the SDAR parameters
        μ̂^j, Ĉ_0^j, Ĉ_1^j, ρ̂_1^j, and σ̂^2 j;
20)   j = j+1;
21) calculate the first value of Z_t
22) return μ̂^j, Ĉ_0^j, Ĉ_1^j, ρ̂_1^j, σ̂^2 j and Z_t.
```

Figure 4.2 - Pseudo-code for the learning phase algorithm.

To update the initial values, the node samples *T* additional observations and sends the first of them to the base station (line 10-13). The SDAR algorithm updates its parameters according to the expressions from (4.7) to (4.11) and stores the results (lines 14-15). The node collects the remaining (*T*-1) values and runs the SDAR algorithm (lines

16-20). Then, the node calculates the first value for Z, Z_T, using the expression in (4.14).

The learning algorithm returns SDAR parameters and the first value of Z_t.

4.5.2 The operation phase

After the learning phase, the node has all the parameters it needs to start the operation phase: the user-set values *(r*, α and *T)*, the SDAR parameters and the first value for Z_t, $t = N_{ini} + T$. Figures 4.3 and 4.4 presents the pseudo-code for TS-SOUND operation phase and post-monitoring algorithm, respectively.

The operation phase continues while the node's battery has a noncritical level of energy (*energy.OK*=1). The node reads the sensed value, stores only the last T sensed values (lines 3-5), runs the SDAR algorithm and stores the T+1 last values of the distribution parameters (lines 6-7), and calculates the value of Z_t (line 8).

If the suppression scheme considers that X_t has a small probability to be generated bt the current distribution ($Z_t > Z_T^{\alpha}$), TS-SOUND opens a monitoring window of size T (lines 9-10). During this time interval (Figure 4.4), the node collects data, updates the SDAR parameters and keep their (2T+1) last values. After closing the monitoring window, the node calculates Z_{t+T}^{B} and Z_{t+T}^{A} (line 11) and compares their values with their respective thresholds (line 12). If the outlier detected at time period t is considered to be a change-point, the node summarizes the values collected inside the post-monitoring window using the median and sends this summary to the base station (lines 13-14). Otherwise, since the detected outlier is classified as an aberrant reading, the updates for the SDAR parameters calculated during the monitoring window are replaced by the updates at *t-1*, the time period before the occurrence of the detected outlier (lines 15-16). This procedure avoids the bad effect of aberrant readings on the estimation of the distribution parameters.

TS-SOUND operation.phase()

Input r, T, z_T^{α}, $z_T^{0.05\alpha}$, $\hat{\mu}^t, \hat{C}_0^t, \hat{C}_1^t, \hat{\rho}_1^t, \hat{\sigma}^{2\,t}$, Z_t.

Output *values sent to the base station*

1) $t = (N_{ini} + T)+1$; *# time counter*
2) **every** t_s *time units* **while** (*energy.OK* = 1) **do**
3) **read** x_t ;
4) **enqueue** $X = X \cup x_t$;
5) **keep** *the last T values of X* ;
6) **calculate and store** *SDAR parameters* $\hat{\mu}^t, \hat{C}_0^t, \hat{C}_1^t, \hat{\rho}_1^t, \hat{\sigma}^{2\,t}$;
7) **keep** *the last (T+1) values of* $\hat{\mu}^t, \hat{C}_0^t, \hat{C}_1^t, \hat{\rho}_1^t, \hat{\sigma}^{2\,t}$;
8) **calculate** Z_t ;
9) **if** ($Z_t > z_T^{\alpha}$) **do** *# if an outlier is detected...*
10) **run** monitoring.window() ; *# ... it opens the monitoring window*
11) **calculate** Z_{t+T}^B **and** Z_{t+T}^A ;
12) **if** $Z_{t+T}^B \geq z_T^{\alpha}$ **and** $Z_{t+T}^A \leq z_T^{0.05\alpha}$ **do**
13) **calculate** $\tilde{x}$ = median[$X_{t+1} \ldots X_{t+T}$] ;
14) **send** $\tilde{x}$.
15) **else do**
16) $\left[\hat{\mu}^j, \hat{C}_0^j, \hat{C}_1^j, \hat{\rho}_1^j, \hat{\sigma}^{2\,j}\right]_{j=\{t,t+1,\ldots,t+T\}} = \hat{\mu}^{t-1}, \hat{C}_0^{t-1}, \hat{C}_1^{t-1}, \hat{\rho}_1^{t-1}, \hat{\sigma}^{2\,t-1}$.
17) $t = t + 1$.
18) **send** (x_t , *end.flag*). *# End of node's operation*

Figure 4.3 - Pseudo-code for the TS-SOUND operation phase algorithm

When the node is running out of energy (*energy.OK*=0), the algorithm transmits the last sensed value and an end flag.

Opening a time window after the outlier detection introduces a delay of T time periods in the base station updating. However, we have three reasons to adopt this post-monitoring window. First, it allows for comparing the time series before and after the detected outlier. Second, it allows for summarizing the values generated by the new

distribution. This summary estimates better the next data to be suppressed than the value that was responsible by the alarm raising. Third, it avoids sending the observation detected as an outlier to the base station, since TS-SOUND may mistake an aberrant point for a change-point.

monitoring.window()

Input r, T, *the last (T+1) values of* $\hat{\mu}^t, \hat{C}_0^t, \hat{C}_1^t, \hat{\rho}_1^t, \hat{\sigma}^{2\,t}$.

Output X , *the last (2T+1) values of* $\hat{\mu}^t, \hat{C}_0^t, \hat{C}_1^t, \hat{\rho}_1^t, \hat{\sigma}^{2\,t}$

1b) $j = 1$;
2b) **every** t_s *time units* **while** $(j <= T)$ **do** `# monitoring window`
3b) $t = t + j$;
4b) **read** x_t ;
5b) **enqueue** $X = X \cup x_t$;
6b) **keep** *the last T values of* X ;
7b) **calculate and store** $\hat{\mu}^t, \hat{C}_0^t, \hat{C}_1^t, \hat{\rho}_1^t, \hat{\sigma}^{2\,t}$
8b) **keep** *the last (2T+1) values of* $\hat{\mu}^t, \hat{C}_0^t, \hat{C}_1^t, \hat{\rho}_1^t, \hat{\sigma}^{2\,t}$;
9b) $j = j+1$.
10b) **return** X , *the last (2T+1) values of* $\hat{\mu}^t, \hat{C}_0^t, \hat{C}_1^t, \hat{\rho}_1^t, \hat{\sigma}^{2\,t}$.

Figure 4.4 - Pseudo-code for the post-monitoring window algorithm

4.5.3 Costs

At the end of the learning phase, the node stores N_{ini} values. After that, at each time period t, the node has to store the last T updates for the SDAR parameters ($5T$ values) and the last T sensed values. Besides, the node has to store five user-set parameters. Four of them are permanent *(r,* z_T^{α} , $z_T^{0.05\alpha}$ and T*)*. The size of the learning sample (N_{ini}) can be deleted after the learning phase, as well as the learning sample. During a monitoring window, the node has to store the last ($2T+1$) values of the SDAR parameters, that is, $5(2T+1)$ values. Then, during the

operation phase, the node has to store ($6T+4$) values outside the monitoring window and ($10T+9$) values during the monitoring window.

TS-SOUND operation phase involves mainly simple calculations, as additions and multiplications. The most costly operation is the square-root in the expression $\hat{\sigma}^t = \sqrt{\hat{\sigma}^{2^t}}$. One alternative to decrease the calculation costs is to use the mean absolute deviation (MAD) instead of σ^t to define Z_t. This would elimate the square-root operation. We have run experiments using this alternative. The results are discussed in section 4.7.3.

The message the node sends to the base station contains only the median the data collected during the post-monitoring window.

4.5.4 TS-SOUND as a suppression scheme

In this section, we frame the TS-SOUND protocol as a suppression scheme according to framework proposed in (SILBERSTEIN *et al.*, 2007a). At each time period *t*, the node collects data x_t, updates the SDAR parameters, calculates Z_t and evaluates the function `Z.fcn`, defined as following

$$\texttt{Z.fcn} = \begin{cases} 1, & \text{if } Z_t > z_T^{\alpha} \\ 0, & \text{otherwise} \end{cases} . \qquad (4.18)$$

As in PAQ and EXP schemes, `Z.fcn` evaluates the error between real and predicted values. However, in TS-SOUND case, the calculations of the predicted values are based on a time series model updated at each new sensor reading.

If `Z.fcn` = 1, the nodes opens a monitoring window and, for T time periods, sense and store the data. At time period t+T, the node evaluate two functions, `Zb.fcn` and `Za.fcn`, defined as following

$$\texttt{Zb.fcn} = \begin{cases} 1, & \text{if } Z^B_{t+T} \geq z^{\alpha}_T \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad \texttt{Za.fcn} = \begin{cases} 1, & \text{if } Z^A_{t+T} \leq z^{0.05\alpha}_T \\ 0, & \text{otherwise} \end{cases}, \tag{4.19}$$

where Z^B_{t+T} and Z^A_{t+T} are defined by the expressions (4.15) in and (4.16), respectively. The functions `Z.fcn`, `Zb.fcn` and `Za.fcn` play the role of the g function in the data suppression framework in (SILBERSTEIN *et al.*, 2007a).

To decide if a message has to be sent to the base station, the node uses the following encoding function

$$f_{\text{enc}} = \begin{cases} \tilde{x}_T, & \text{if } (\texttt{Zb.fcn} \cap \texttt{Za.fcn}) \\ \perp, & \text{otherwise} \end{cases}, \tag{4.20}$$

where $\tilde{x}_T$ is the median of the T values read inside the post-monitoring window. If $T=1$, $\tilde{x}_1 = x_{t+1}$.

At each time period t, the base station waits for the report r_t from the nodes and uses the following decoder function to update its database

$$\hat{x}_t = \begin{cases} \tilde{x}_T, & \text{if } r_t = \tilde{x}_T \\ \hat{x}_{(t-1)}, & \text{if } r_t = \perp \end{cases}. \tag{4.21}$$

VB and TS-SOUND schemes have similar encoding and decoding functions. They send only one value to the base station. In case of data suppression, the last sent value is the estimative for the current time period.

4.5.5 On TS-SOUND's parameters

TS-SOUND scheme is defined by three parameters: the size of the time windows *(T)*; the amount of change in the expected behavior of the monitored variable we want to

detect (α) and how much weight the current observation must have in the on-line updating of the distribution parameters (r).

As the length of the post-monitoring, the value of T should be as large as the size of the sequence of aberrant readings. On the other hand, we should choose a small value for T to decrease the delays to detect an outlier and to update the base station if a change-point occurs.

As we will discuss in section 4.7, we do not know how large the clusters of aberrant readings will be. Then, the choice of the value for T must consider TS-SOUND's performance when it is applied on time series with sequences of aberrant readings of several sizes. Then, we have to choose the value of T that produces the most homogeneous performances considering aberrant clusters of different sizes. The experiments results in section 4.7 will help us to make this choice.

On choosing the value of r, we should consider how large the local variation of time series is. For instance, a wind speed time series has a local variation larger than the local variation of an atmospheric pressure time series (Figure 4.5, section 4.6). Therefore, the current observation in a wind speed series should have a weight (r) larger than the weight of the current observation in an atmospheric pressure series. However, giving larger weights to the observation in the estimation of the distribution parameters makes harder to detect this observation as an outlier. In fact, as we will see in section 4.7, values for r larger than 0.1 have degraded the suppression rates in the evaluation experiments.

The value of α is the probability of making a mistake: detecting a non-outlier as an outlier. If we set a small value for α, we decrease this error probability. However, small values for α make harder the detection of change-points, especially if these points represent a small change in the expected behavior of the time series. On the other hand, if α is large, the scheme will be able to detect small changes, even though false

outlier alarms will rise more often. Then, the user has to define what is more important to him/her: capturing small changes or avoiding aberrant readings.

4.6 Evaluation Experiments

In this section, we describe a set of extensive experiments to evaluate the performance of the TS-SOUND suppression scheme.

4.6.1 The data

We have used real data collected by the weather station of the University of Washington (USA)[20]. Our goal has been to account for diverse types of temporal behavior. Then, we have selected time series for wind speed (nautical miles per hour), air temperature (F), air relative humidity (%) and atmospheric pressure (milibars). The temporal resolution is one measurement per minute (average of measurements at each 5 seconds). To account for seasonal variability in the weather data, we have chosen four different months (October'06, January'07, April'07 and July'07). For each month, we have selected the data of the days from 10^{th} to 16^{th}. We have run the experiments using these 28 daily time series (1440 readings per series) for each variable.

Figure 4.5 presents the typical daily time series for each variable. These time series present different behaviors: from series with large local movements relative to its global variation (wind speed) until series with small local movements relative to its global variation (atmospheric pressure).

[20] http://www-k12.atmos.washington.edu/k12/grayskies/nw_weather.html

4.6.2 The experiments

We have designed the experiments to evaluate the performance of TS-SOUND scheme and compare it with the performance of the following suppression schemes: value-based (VB), exponential regression (EXP) and PAQ.

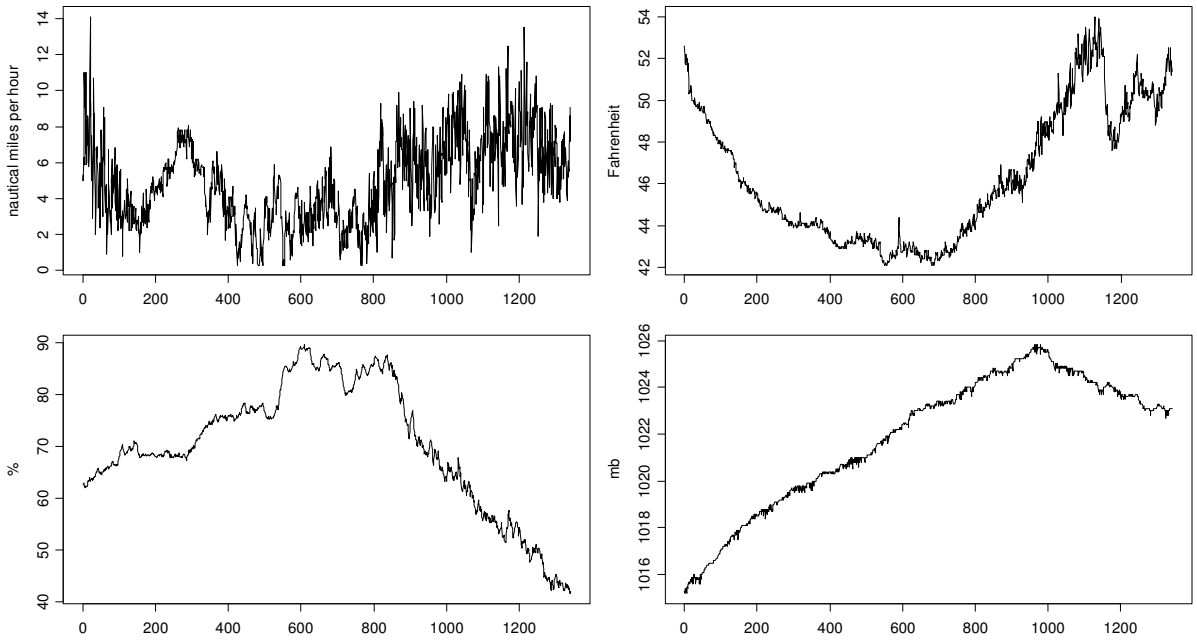


Figure 4.5 - Typical daily time series used in the evaluation experiments.
From left to right: wind speed (July'07); air temperature (April'06); air relative humidity (October'06); atmospheric pressure (April'07).

For the parameters of TS-SOUND scheme, we have set the values r = (0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6), α = (0.25, 0.20, 0.15, 0.10, 0.05, 0.025, 0.01) and T = (2, 4, 6, 8, 10). The value for the threshold z_T^{α} corresponds to the percentile 100(1-α/2) of the Gaussian distribution with zero mean and standard deviation $\sqrt{T}$. The first *(100 + T)* values of the time series composed the learning sample. We have run the experiments using the R environment (R DEVELOPMENT CORE TEAM, 2005).

Making the TS-SOUND scheme comparable to the other evaluated schemes (PAQ, EXP

and VB) is not a trivial task, since they use different criteria to trigger their data sending. The latter schemes use absolute value of the prediction error to decide when the node must send data to the base station, whereas TS-SOUND uses the detection/classification of outliers. Then, we have had to answer the question: "how to choose values for ε_υ and ε_{VB} (PAQ/EXP and VB error thresholds, respectively) so that we make these schemes comparable to TS-SOUND scheme using the values chosen for α ?"

Our solution for this problem has been to use the prediction errors of TS-SOUND scheme to define the values for ε_υ and ε_{VB} . Then, after applying the TS-SOUND scheme to a real time series data using a given value for α, we have calculated the absolute prediction error as following

$$AE_t = \left| x_t - \hat{x}_t \right|, \quad t = 1, 2, ..., N_{TS} \tag{4.22}$$

where $\hat{x}_t$ is prediction value for real data x_t and N_{TS} is the size of the time series. To avoid the influence of discrepant values, we have decided to throw out the 10% largest values of AE_t and define the value for ε_υ and ε_{VB} as the percentile 90 of the AE_t values. Therefore, the maximum error of the predictions using PAQ, EXP and VB schemes is the percentile 90 of the prediction error of TS-SOUND schemes. Once the range of the absolute prediction error has been equalized, the distribution of the error within this range will be determined by the performance of the evaluated schemes.

The values for the other parameters of PAQ and EXP have been chosen based on the values cited in (TULONE and MADDEN, 2006) as good choices: $\varepsilon_\delta = (1.8/3.0)\varepsilon_\upsilon$, $A_{PAQ}=(5, 15)$ and $a=(8/15)A_{PAQ}$. The learning sample size (N_{LS}) has been set as the first 100 observations of the time series.

4.6.2.1 Evaluating the influence of aberrant readings

We have designed an experiment to evaluate how sensitive to aberrant points are the

suppression schemes analyzed in this chapter. This experiment has used the real time series previously described. For each time series, we have inserted aberrant values, isolated or clustered, in randomly chosen time periods. To generate isolated aberrant readings, we have sampled 100 time periods of a given time series to be replaced by an aberrant reading, preserving a minimum interval of 11 time periods between two sequential positions. Then, about 10% of a time series has been composed by aberrant points. To generate the aberrant reading at the selected time period, we have used the interquartile range IQ, defined as $IQ = P_{diff}(75) - P_{diff}(25)$, where $P_{diff}(p)$ is the percentile p of the sequential differences $|X_t - X_{t-1}|$. In a boxplot analysis (TUKEY, 1977), values smaller than $P_{diff}(25) - 3 \times IQ$ or greater than $P_{diff}(75) + 3 \times IQ$ are considered to be extreme outliers. Then, to generate an aberrant reading, we have added ($sign \times range \times IQ$) to the current value of the candidate time period, where range has been randomly chosen inside the interval [3 ; 6] and sign has been randomly chosen between -1 and +1. Adopting the boxplot's rule and a random value for range, we have expected to decrease our influence on the generation of the aberrant values.

In addition to isolated aberrant readings, we have generated sequences with 2, 3, 4 and 5 aberrant readings. From now on, we will denote the sequences of aberrant readings by *aberrant sequences*. To produce such sequences, we have supposed that all the aberrant readings in a cluster are generated in a same direction, as those ones presented in Figure 4.1. Given the size of the sequence, we have grouped the initial 100 aberrant readings. For instance, in the experiments with sequences of 4 aberrant points, we have generated 25 sequences. The first reading of the sequence has been inserted in the time series as in the isolated case. To generate the sequential aberrant readings, we have used the same rule to produce the first aberrant reading. However, their signs have been constrained to the sign of the first reading in the cluster. We have applied TS-SOUND, PAQ, EXP and VB schemes on these modified time series using as parameters the values described in the previous section.

4.6.2.2 Assessing the performance of the suppression schemes

We have evaluated the performance of suppression schemes using the trade-off between two measures: the *suppression rate* and the *prediction error*.

We have adopted the *median absolute error* (MAE) to measure the prediction error. The median absolute error has been calculated as

$$MAE = median_{(t=1,2,...,N_{TS})} \left| x_t - \hat{x}_t \right|, \tag{4.23}$$

where N_{TS} is the size of the time series.

We can cite some advantages of adopting MAE to assess the prediction error instead of using other error measures such as the mean square error (MSE). First, the absolute difference between predicted and real values is an intuitive measure for the prediction error. Second, the absolute error preserves the original measurements units, which makes easier its interpretation. Finally, the median is more robust to the influence of discrepant values.

The suppression rate (SR) has been calculated as the proportion of suppressed data

$$\mathrm{SR} = 1\text{-}\frac{(\text{number of sent messages})}{N_{TS}}. \tag{4.24}$$

If a scheme increases its suppression rate, we expect MAE also increases, since the node updates the base station database less often. A suppression scheme S1 can be defined as better than other suppression scheme S2 if, for a given value of prediction error, S1 is able to get suppression rates larger than the suppression rates of S2.

To evaluate the robustness to aberrant readings of TS-SOUND scheme, we have

calculated the odds of sending data to the base station provided that an aberrant reading has been detected as

$$Odds_{SENT}^{Aberrant} = \frac{\text{number of detected aberrant readings that have caused data sending}}{\text{number of detected aberrant readings that have not caused data sending}}. \quad (4.25)$$

A TS-SOUND scheme is considered to be robust to aberrant readings if its $Odds_{SENT}^{Aberrant}$ is smaller than 1. Then, a suppression scheme S1 can be defined as more robust to aberrant readings than a suppression scheme S2 if S1 has got an odds of sending data smaller than S2's odds.

Since PAQ, EXP and VB schemes always send the detected outliers to the base station, their $Odds_{SENT}^{Aberrant}$ are infinite. Then, we have evaluated the robustness to aberrant readings of these schemes by comparing their suppression rates in the time series with and without aberrant readings. For a robust scheme, this ratio is close to 1.

4.7 The results

In this section, we present the main results of the experiments described in the previous section. We start our analysis with a simple case study.

4.7.1 A simple case study

We have had access to the air temperature and relative humidity data collected by three Tmote Sky sensor nodes[21]. They have collected data at each 30 seconds during 32

[21] Thanks to the Professor Rone Ilídio da Silva of Universidade Presidente Antônio Carlos (Campus Conselheiro Lafaiete), for making these data available.

hours. Each sensor node has produced about 4000 readings of each variable. The top of the Figure 4.6 presents the time plot of the temperature data collected by the sensor node 2.

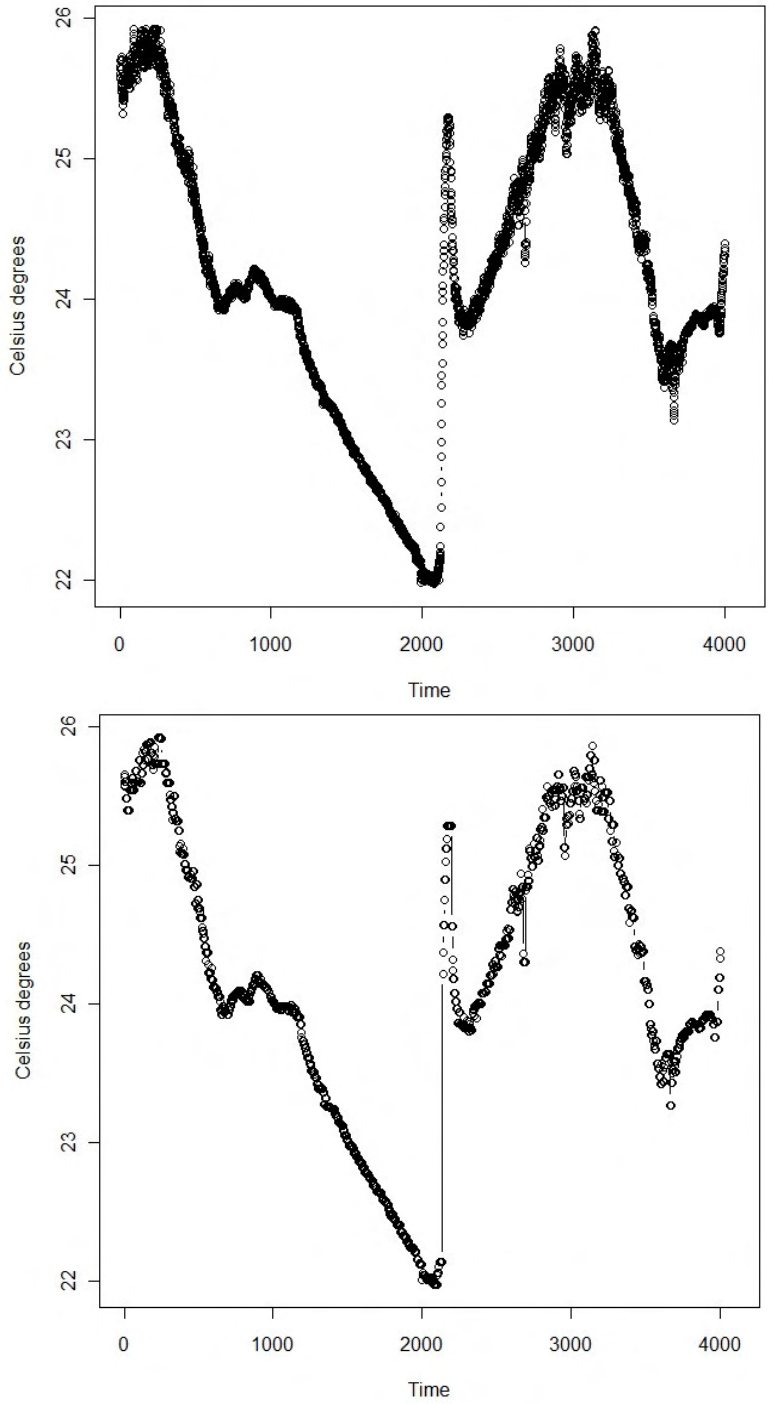


Figure 4.6 - Results of TS-SOUND scheme applied to data collected by Tmote Sky.
At the top: time series predicted at the base station.
At the bottom: real air temperature data collected at the sensor node 2.

Since these data have not enough time series to be used in an extensive evaluation, we have used them to perform an initial analysis. Table 4.1 presents the values for the performance measures of the evaluated schemes using T=2, α=0.15, r=0.1 (TS-SOUND's parameters) and A_{PAQ} = 15 (PAQ and EXP's parameter). The values for ε_{υ} and ε_{VB} were determined as we have described in section 4.6.2.

For both variables, TS-SOUND has got suppression rates similar to the rates of the other schemes, whereas its prediction error has been smaller than the prediction error of the other schemes. The top of the Figure 4.6 presents the time series predicted at the base station when the TS-SOUND scheme has been applied to the temperature data collected by the sensor node 2. Comparing the real and predicted series, we have noticed that TS-SOUND avoids reporting the erratic movement of the series as, for instance, in the beginning and final parts of the time series in the Figure 4.6. On one hand, TS-SOUND delays the notification of fast changes such as the one near the time period 2000. TS-SOUND classifies this behavior as an aberrant one until it notices there is a change. From this moment on, it updates the base station more often. On the other hand, likely clusters of aberrant readings are represented by few updates, as those ones near the time period 3000.

Since no messages can be sent to base station during the TS-SOUND's monitoring window, increasing its size *(T)* has increased the suppression rates. As a result, the value of the median absolute error has also increased. The parameter α has had a similar effect on the suppression rates and prediction errors: the larger the rigor to consider an observation as an outlier, the larger the chance of suppressing data.

On the value of r, our initial experiments have pointed to r=0.1 as the value that produces the best trade-off between the suppression rate and the prediction error. This means that we obtain the best performance for TS-SOUND when the on-line estimation of the new values for the distribution parameters sets less weight to the current sensor reading (equations (4.7) to (4.11)). TS-SOUND schemes using r

values smaller than 0.1 have produced results very similar to the results with r=0.1. However, increasing the value of r up to 0.5 has degraded the suppression rates. In fact, giving larger weights to the observation in the estimation of the distribution parameters makes harder to detect this observation as an outlier.

Table 4.1 - Results of the evaluation experiments applied to data collected by three Tmote Sky sensor nodes. Air Temperature (°C) and Relative Humidity (%) data. Suppression rate and median absolute error are within the parenthesis.

Scheme	Sensor Node 1		Sensor Node 2		Sensor Node 3	
	Temperature ($\varepsilon_\upsilon = \varepsilon_{VB}$ = 0.03 °C)	Relative Humidity ($\varepsilon_\upsilon = \varepsilon_{VB}$ =0.41 %)	Temperature ($\varepsilon_\upsilon = \varepsilon_{VB}$ = 0.08 °C)	Relative Humidity ($\varepsilon_\upsilon = \varepsilon_{VB}$ = 0.25%)	Temperature ($\varepsilon_\upsilon = \varepsilon_{VB}$= 0.03 °C)	Relative Humidity ($\varepsilon_\upsilon = \varepsilon_{VB}$ = 0.17%)
TS-SOUND (r=0.1;T=2; α =0.15)	0.823 (0.005 °C)	0.857 (0.057 %)	0.858 (0.015 °C)	0.877 (0.040 %)	0.865 (0.010 °C)	0.883 (0.020 %)
PAQ (A_{PAQ}=15)	0.753 (0.014 °C)	0.807 (0.157 %)	0.812 (0.031 °C)	0.826 (0.086 %)	0.783 (0.010 °C)	0.836 (0.053 %)
EXP (A_{PAQ}=15)	0.893 (0.010 °C)	0.816 (0.146 %)	0.825 (0.028 °C)	0.829 (0.082 %)	0.789 (0.009 °C)	0.846 (0.051 %)
VB	0.858 (0.010 °C)	0.872 (0.124 %)	0.874 (0.020 °C)	0.892 (0.086 %)	0.859 (0.010 °C)	0.897 (0.041 %)

4.7.2 Selecting the best value for the length of the monitoring window

TS-SOUND's strategy to distinguish a change-point from an aberrant reading is to use a post-monitoring window whenever an outlier is detected. This time window works as a filter of aberrant readings and makes TS-SOUND robust to these erroneous data. The success of this filtering strategy is closely related to the length of the monitoring window. We expect large aberrant sequences require large

windows to be filtered. However, we do not know how large the clusters of aberrant readings will be.

In this section, we examine the results of experiments with the meteorological data of the University of Washington to answer the following question: "Considering several sizes for the clusters of aberrant readings, which is the minimum value for the length of the monitoring window that leads to TS-SOUND scheme with

a) the largest robustness to aberrant readings and
b) the best trade-off between suppression rate and prediction error ?"

To answer the first part of the question, we have summarized some of the experiments results using plots as the ones in figures 4.7 and 4.8. They present the odds of "sending data to the base station provided that an aberrant reading has been detected" as a function of the length of the monitoring window considering aberrant sequences of several sizes. Figures 4.7 and 4.8 present the results for the sets of time series that have got the most irregular behaviors: wind speed and air relative humidity measurements, respectively. We have looked for the smallest length for the monitoring window that leads to the most similar values for the odds among aberrant clusters of different sizes. For the wind speed time series, the monitoring windows of length 10 and 2 have presented the most similar odds. Then, the chosen length is $T=2$. For the air relative humidity, the length is also $T=2$. For air temperature and atmospheric pressure time series, the larger the monitoring window is, the less similar the odds are. Therefore, $T=2$ is the chosen length.

Increasing the value of α decreases the odds of "sending data to the base station provided that an aberrant reading has been detected", since the rigor to classify an observation as an outlier increases.

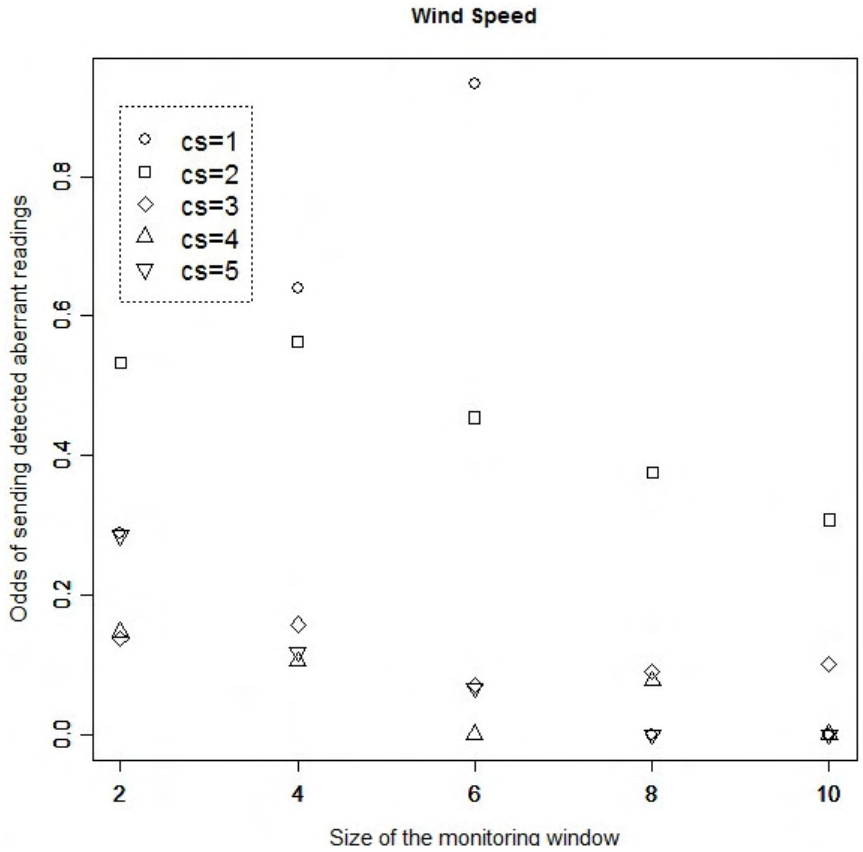


Figure 4.7 - Robustness to aberrant readings of TS-SOUND scheme.
According to the length of the monitoring window *(T)* and the size of the aberrant clusters (CS). The other TS-SOUND's parameters have been $\alpha = 0.15$ and $r = 0.1$.

We have answered the second part of the question by examining plots as the ones in figures 4.9 and 4.10. They present the trade-off between suppression rate and prediction error for several lengths of the monitoring window and considering aberrant sequences of different sizes. We have looked for the smallest length for the monitoring window that leads to the most similar suppression rates and prediction errors among aberrant clusters of different sizes. In figures 4.9 and 4.10, we have looked for the group of symbols (*T* values) that are more "clustered". For wind speed and air relative humidity time series (figures 4.9 and 4.10, respectively), the monitoring windows of length 6 and 4 have presented the most similar suppression rates and prediction errors.

Then, the chosen length is $T=4$. Examining the air temperature and atmospheric pressure time series, we have got the same value for T.

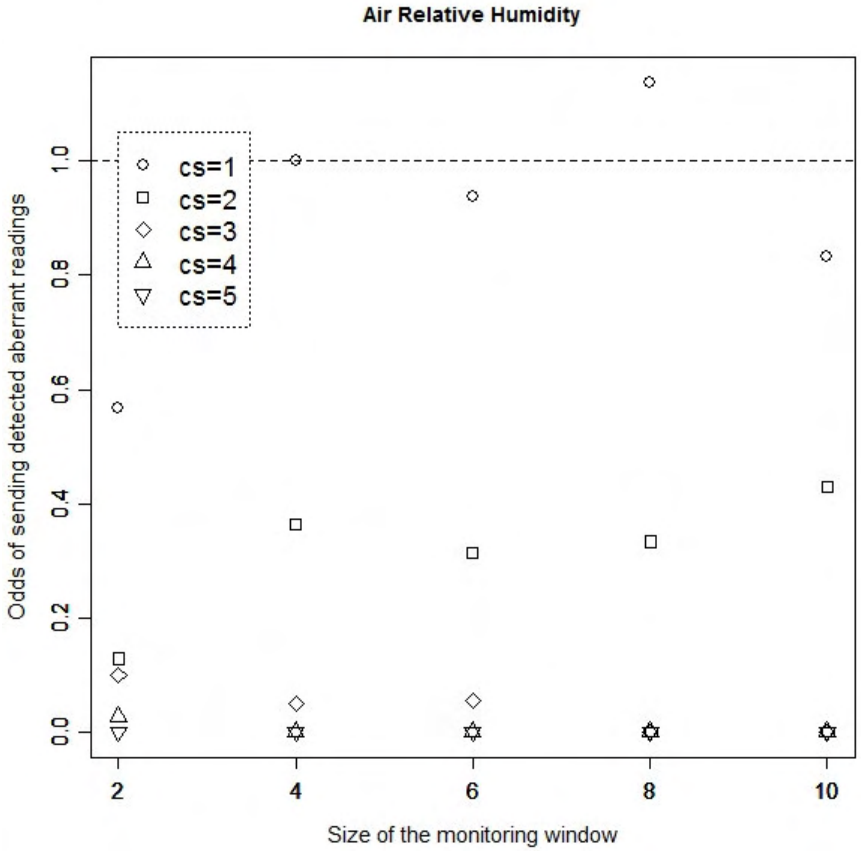


Figure 4.8 - Robustness to aberrant readings of TS-SOUND scheme.
According to the length of the monitoring window *(T)* and the size of the aberrant clusters (CS). The other TS-SOUND's parameters have been $\alpha=0.15$ and $r=0.1$.

Since we have got different answers for the two parts of the proposed question, we have chosen the best value for T by examining the effect of using the value chosen in part (a) on the context of part (b) and vice versa. Then, we have examined the effect of choosing $T=2$ on the trade-off between suppression rate and prediction error and the effect of using $T=4$ on the odds of "sending data to the base station provided that an aberrant reading has been detected". In the former case, exchanging $T=4$ for $T=2$ produces a substantial increasing in the dissimilarity of

the suppression rates and prediction errors for the wind speed, air temperature and atmospheric pressure time series. In the latter case, the effect of exchanging the values of T (T=2 for T=4) is smaller than in the former case. The worst effect has occurred in the air relative humidity time series (Figure 4.8). For T=4, the odds of "sending data to the base station provided that an aberrant reading has been detected" is, in median, equal to 1 when isolated aberrant readings (CS=1) occur in the time series. However, the other odds are smaller than 1. Then, considering all evaluated time series and sizes for aberrant clusters, we have chosen the value 4 as the best one for the length of the monitoring window.

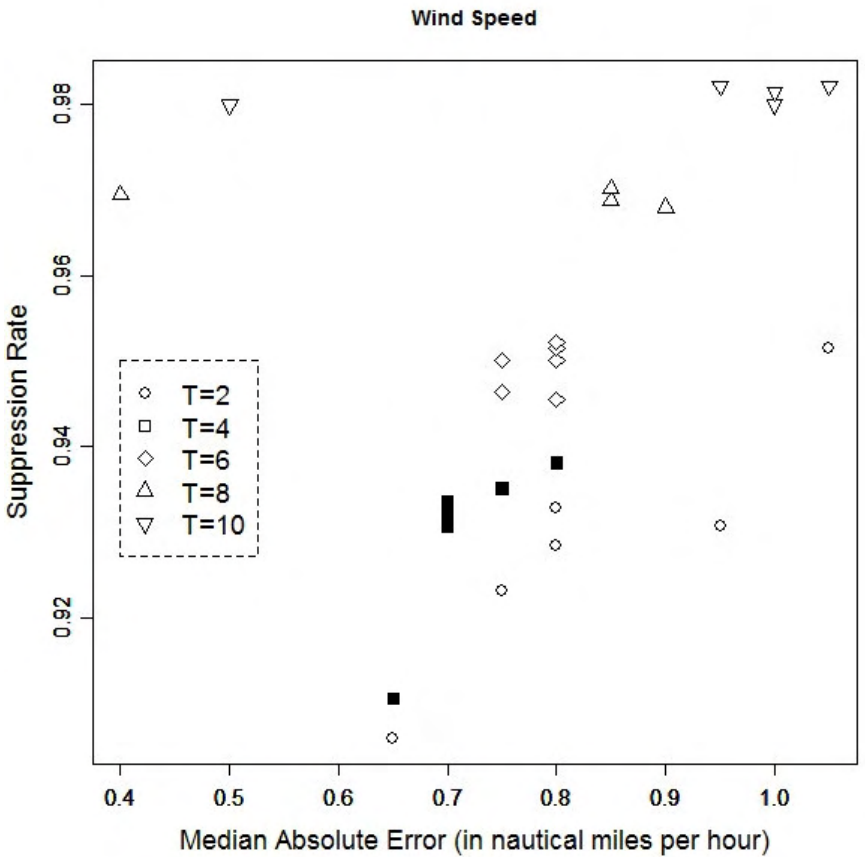


Figure 4.9 - Performance of TS-SOUND scheme applied to wind speed time series. The parameters have been α=0.15, r=0.1 and several values for the length of the monitoring window *(T)*. Each point represents the summary of the results for time series with aberrant clusters of different sizes: 0 (no aberrant readings), 1 (isolated aberrant readings), 2, 3, 4 and 5.

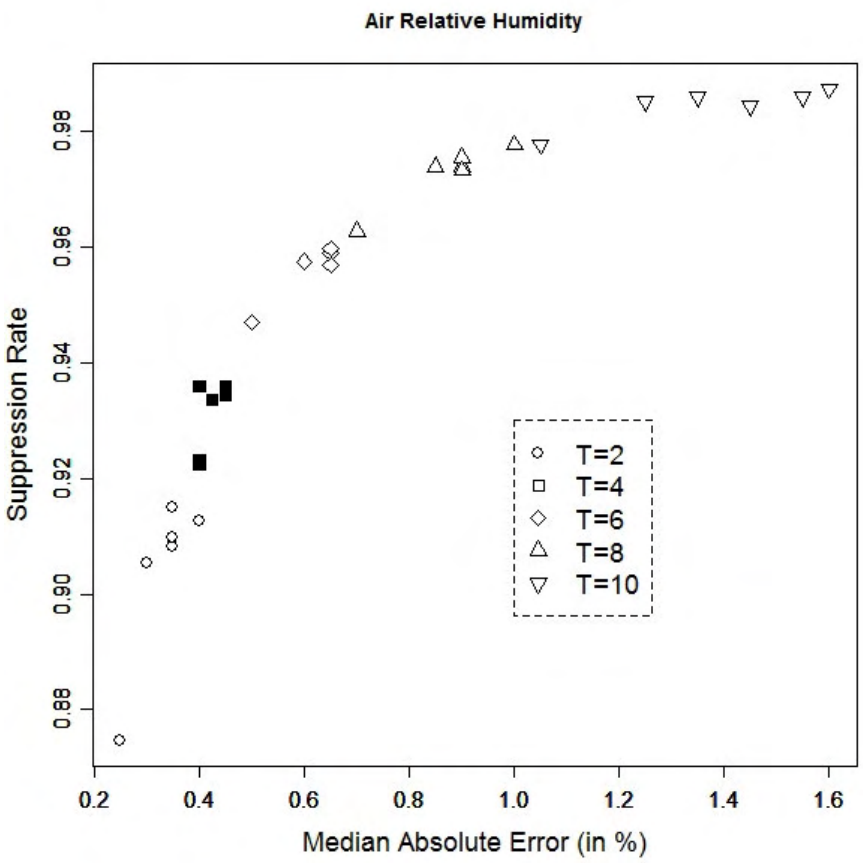


Figure 4.10 - Performance of TS-SOUND scheme applied to relative humidity time series. The parameters have been $\alpha=0.15$, $r=0.1$ and several values for the length of the monitoring window *(T)*. Each point represents the summary of the results for time series with aberrant clusters of different sizes: 0 (no aberrant readings), 1 (isolated aberrant readings), 2, 3, 4 and 5.

4.7.3 Evaluating the schemes' performances

In this section, we compare the performance of TS-SOUND scheme using $T= 4$, selected in previous section, with the performance of PAQ, EXP and VB schemes.

As we have mentioned in section 4.6, we have used the trade-off between the suppression rate and the prediction error of a scheme as a measure for its performance. We represent graphically this trade-off for each one of the sets of meteorological time series using the scatter plots of the figures from 4.11 to 4.14. Each point of a scheme represents the summary of its performance using a different value for α (0.15, 0.10, 0.05, 0.025, 0.01), in this order, following the increasing of the suppression rates. For PAQ/EXP and VB schemes, the values for the correspondent error thresholds ε_{υ} and $\boldsymbol{\varepsilon}_{VB}$, respectively, have been defined as described in section 4.6.2. Points closer to the upper-left corner represent the schemes with the best performances. Since TS-SOUND with T=4 has got its worst results when the time series had isolated aberrant readings (figures 4.6 and 4.7), we have chosen this scenario to compare TS-SOUND's performance with the performance of the other evaluated schemes. The upper and bottom subfigures illustrate which data the base station would have if the node applied TS-SOUND and VB schemes, respectively, on the real time series presented in the middle subfigure. The real time series in the middle subfigures are the original ones in Figure 4.5 with generated aberrant clusters of size 1 (isolated aberrant readings).

To understand what values we should expect for the prediction errors so that we could consider them acceptable, we have used the size of the sequential changes in the time series as a basis for comparison. Then, we have calculated the sequential absolute differences, $|X_t - X_{t-1}|$, in the series of each variable and summarized the sequential changes (non-zero differences) using the percentiles 5 and 95. Therefore, in the air relative humidity and temperature time series, 90% of the sequential changes are within the interval [0.10 ; 1.0]% and [0.10 ; 1.0]F, respectively. In the atmospheric pressure time series, 90% of the sequential changes are within the interval [0.10 ; 0.40] mb. In the wind speed time series, 90% of the sequential changes are within the interval [0.10 ; 2.1] nautical miles. Analyzing figures from 4.11 to 4.14, we notice all evaluated schemes have

got median prediction errors compatible with the expected sequential changes in a given type of meteorological time series. In other words, all evaluated schemes have got acceptable errors on predicting the real time series at base station.

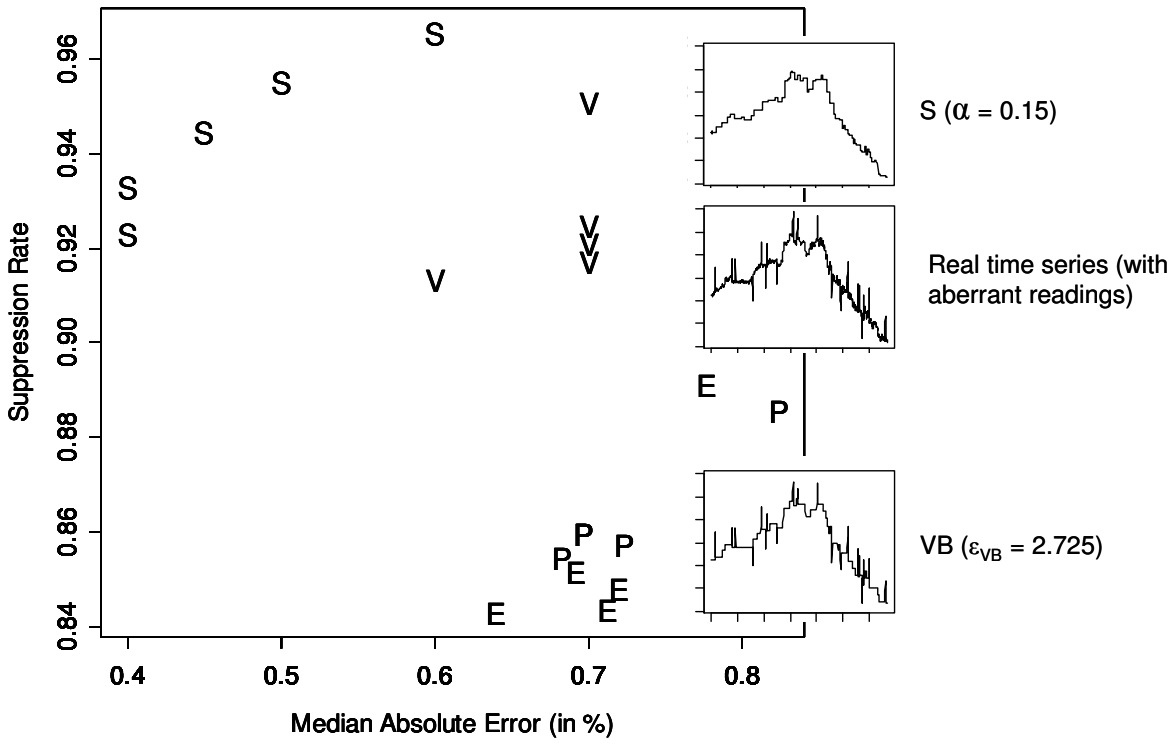


Figure 4.11 - Performance of the evaluated schemes in air relative humidity time series with isolated aberrant readings

Legend: S for TS-SOUND (r=0.1, T = 4), V for value-based, P for PAQ (A_{PAQ}=15) and E for EXP (A_{PAQ}=15). Each point of a scheme represents the summary of its performance using a different value for α (0.15, 0.10, 0.05, 0.025, 0.01), in this order, following the increasing of the suppression rates. For PAQ/EXP and VB schemes, the values for the correspondent error thresholds ε_{υ} and ε_{VB}, respectively, have been defined as described in section 4.6.2.

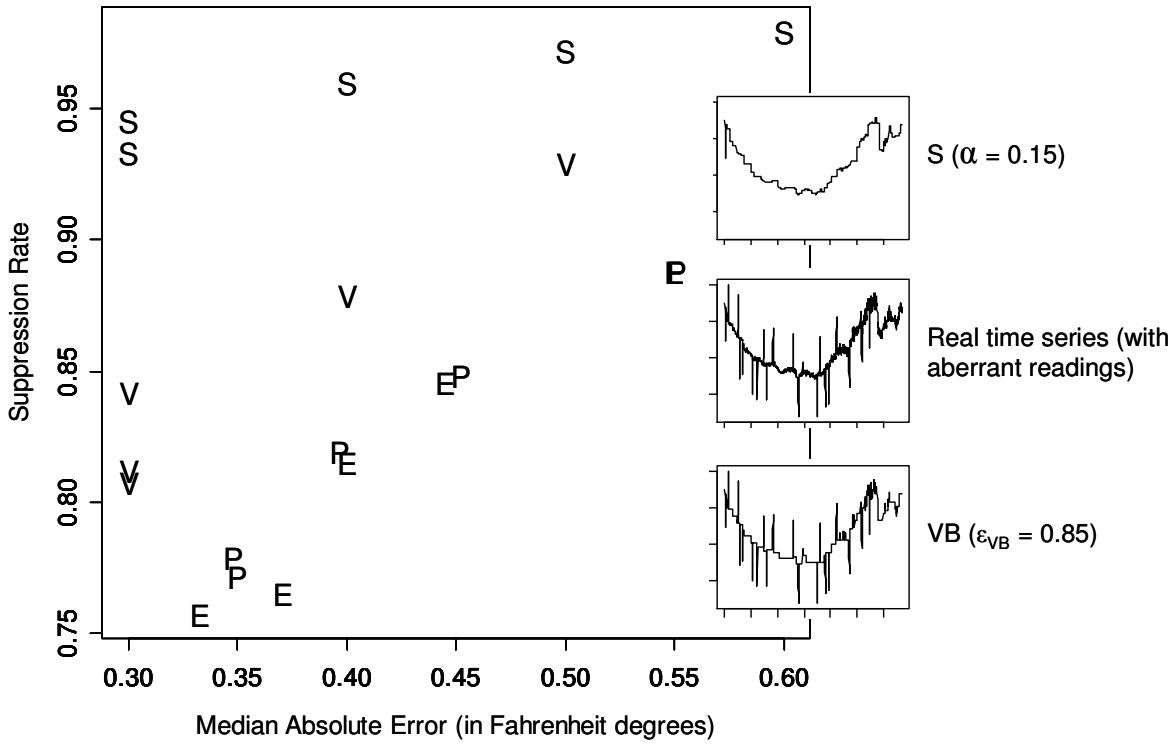


Figure 4.12 - Performance of the evaluated schemes in air temperature time series with isolated aberrant readings. The legend and other details are in the caption of Figure 4.11.

TS-SOUND scheme has got its best performance in air relative humidity and temperature time series (figures 4.11 and 4.12, respectively). In the air relative humidity data, TS-SOUND has been the scheme with the best performance for all values of α, reaching the highest suppression rates and the smallest prediction errors. For the smallest two values of α in the air temperature data and for $\alpha = (0.10,0.05)$ in the atmospheric pressure data (Figure 4.13), the prediction errors of the TS-SOUND and VB are, in median, the same. However, TS-SOUND has got suppression rates higher than VB's rates.

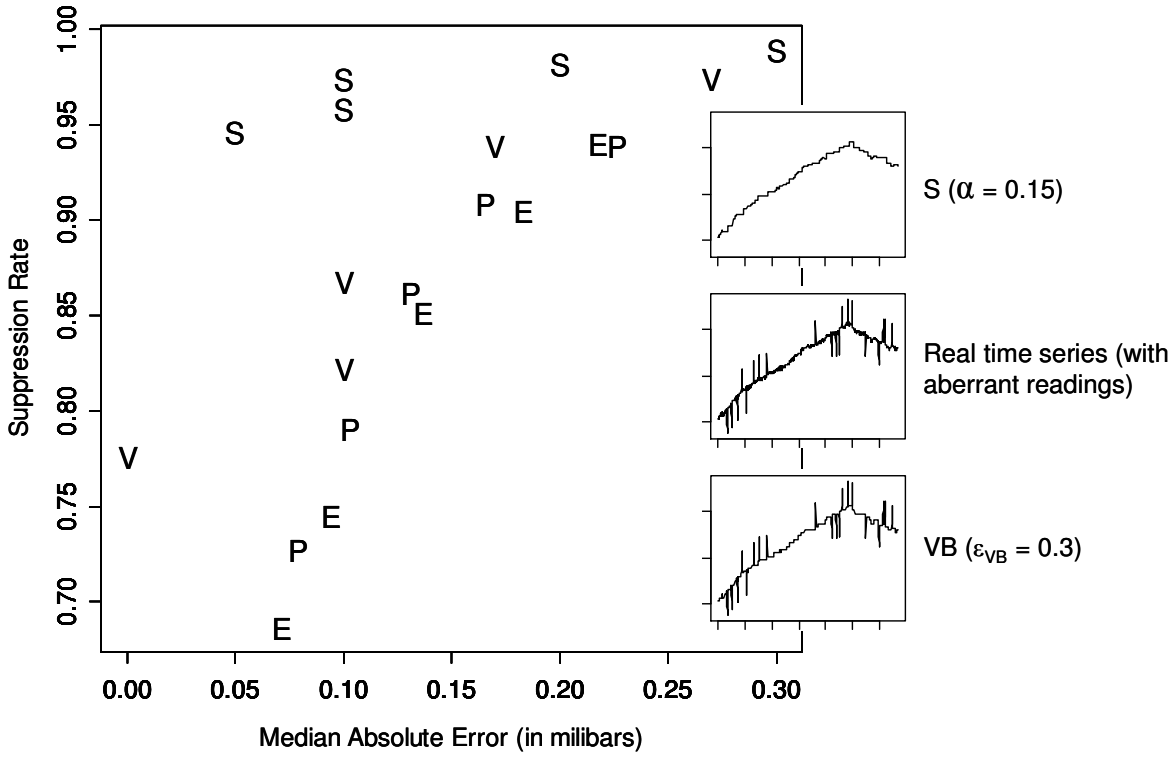


Figure 4.13 - Performance of the evaluated schemes in atmospheric pressure series with isolated aberrant readings. The legend and other details are in the caption of Figure 4.11.

In the wind speed time series, which have a large local variation, TS-SOUND has increased the prediction errors in comparison to the other schemes' errors (Figure 4.14). Nevertheless, it has got a higher increase in the suppression rates in relation to maximum possible increasing. As an example, for α=0.15, TS-SOUND has got a median prediction error of 0.8 nautical miles per hour, which has been 14% larger than VB's median prediction error. However, TS-SOUND's suppression rate has been 0.938, whereas VB has got 0.798. Then, TS-SOUND's rate has got an increasing of 69% in relation to maximum increasing in the VB rate (1 – 0.798). For α=0.10, TS-SOUND's error has been 43% larger than VB's error but TS-SOUND's has increased the suppression rate in 77% of the maximum possible increasing. If we compare TS-SOUND with the PAQ and EXP schemes, the gains are higher.

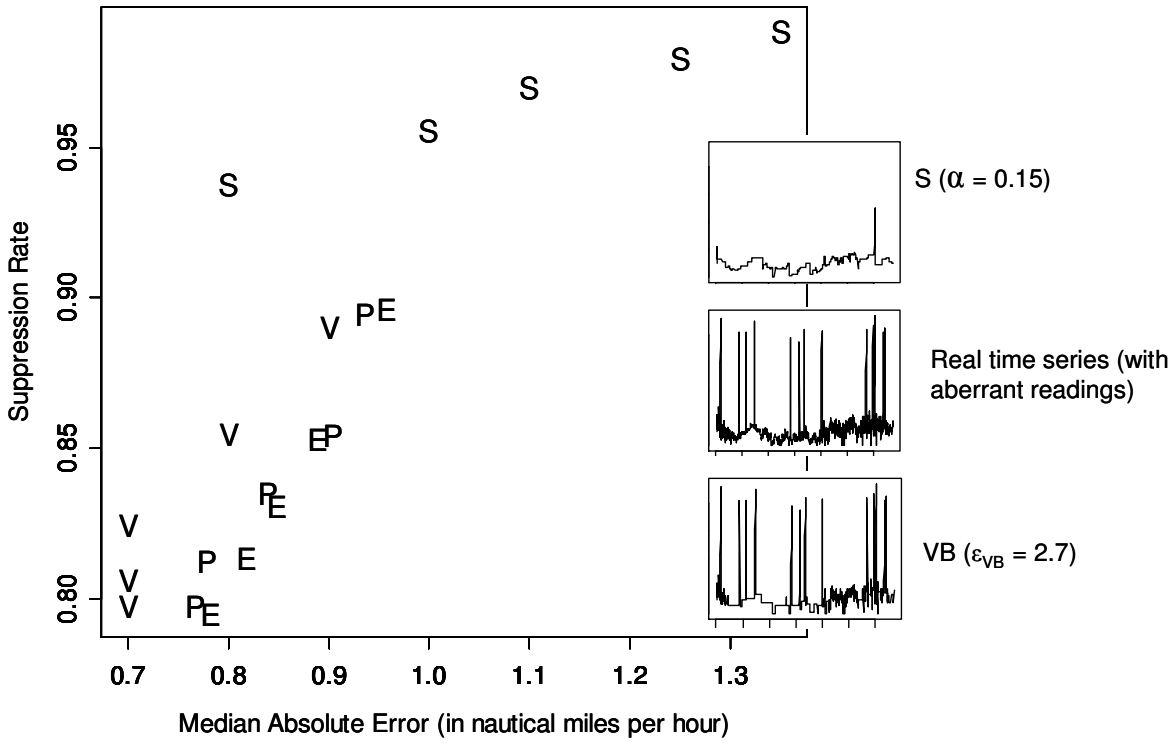


Figure 4.14 - Performance of the evaluated schemes in wind speed time series with isolated aberrant readings. The legend and other details are in the caption of Figure 4.11.

In time series with small local variation, as the atmospheric pressure series, VB scheme has got median prediction errors equal to zero, even suppressing about 77% of the readings (Figure 4.13). However, the correspondent TS-SOUND scheme has suppressed about 95% of the readings, in median, at the cost of increasing 0.05 milibars in the prediction error. Since this increasing is among the 5% smallest sequential changes in atmospheric pressure series, we conclude it is worth to adopt TS-SOUND for this type of data, getting a higher suppression rate at the cost of a small increasing in the prediction error.

On choosing the best value for α, we have to consider how large the local variations in the series are. Comparing figures 4.13, 4.11, 4.12 and 4.14 (in this order), we conclude the larger the local variation the larger the best value for α

must be. In general, for values of α smaller than 0.05, the increasing in the suppression rate does not compensate the increasing in the prediction error.

Comparing the predicted time series to the real ones (subfigures), we notice the robustness to the aberrant readings of TS-SOUND scheme, whereas VB suffers a large influence of these erroneous data. VB's predicted series are similar to the series with aberrant readings (middle subfigures), whereas TS-SOUND's predicted series look like the original series, without aberrant readings, in Figure 4.5.

PAQ and EXP schemes using the largest monitoring window (A_{PAQ}=15) have got suppression rates larger than the rates of those schemes using a smaller window (A_{PAQ}=5). Therefore, PAQ and EXP schemes having a larger period to evaluate the re-estimation of the model parameters have been a better alternative, even if the prediction errors have been slightly larger. Despite of having updated the base station more often than the other schemes, PAQ and EXP schemes have not got the smallest prediction errors. In other words, using these model-based suppression schemes is not a good strategy if the dataset may have aberrant readings.

We have run experiments using the mean absolute deviation (MAD) as a less costly alternative for σ to define Z_t, Z_{t+T}^B and Z_{t+T}^A (expressions 4.14 to 4.16). To update the values of MAD, SDAR algorithm has adapted the expression in (4.11) and used the following expression

$$MAD^t = (1-r)MAD^{t-1} + r\left|X_t - \hat{w}^t\right|. \tag{4.26}$$

Comparing the results of the experiments using both definitions for Z_t, Z_{t+T}^B and Z_{t+T}^A, we have observed that MAD increased the suppression rates and, as a consequence, the prediction errors. This increasing in the errors was especially large for wind speed, air temperature and relative humidity time series when these series had none or small

sequences of aberrant readings (1, 2 or 3 observations). Using MAD instead of σ has made TS-SOUND less sensitive to outliers, which made harder the detection of change-points. On one hand, this can explain the larger predictions errors. On the other hand, "MAD-alternative" has decreased the odds of sending an aberrant reading.

As we notice in figures 4.11 to 4.14, TS-SOUND using σ has got high suppression rates. An alternative that increases these rates at the cost of increasing the prediction errors is not interesting for the network user. Then, we have decided to keep the version of TS-SOUND that uses σ as the measure for data dispersion.

4.7.4 Evaluating the schemes' robustness to aberrant clusters

In this section, we compare the robustness to aberrant clusters of the suppression schemes. Since the $Odds_{SENT}^{Aberrant}$ of PAQ, EXP and VB are infinite, we have calculated the ratio between the suppression rates with and without aberrant clusters. A suppression scheme robust to aberrant readings should present this ratio close to 1. For a suppression scheme that suffers the influence of aberrant readings, this ratio is smaller than 1.

Figures 4.15 and 4.16 present the ratios for the suppression schemes applied on atmospheric pressure and wind speed time series. In these sets of series, the evaluated schemes have suffered the largest and the smallest influence of aberrant clusters, respectively.

The suppression rates of TS-SOUND scheme have not presented relevant changes, whereas the suppression rates of the other schemes have decreased, especially for PAQ and EXP schemes. This is because the model-based prediction adopted by PAQ/EXP schemes is quite sensitive to aberrant readings. They decrease PAQ/EXP's suppression rates for two reasons: the node has to send them as detected outliers to the base station and they cause the re-estimation (and sending) of the new model parameters.

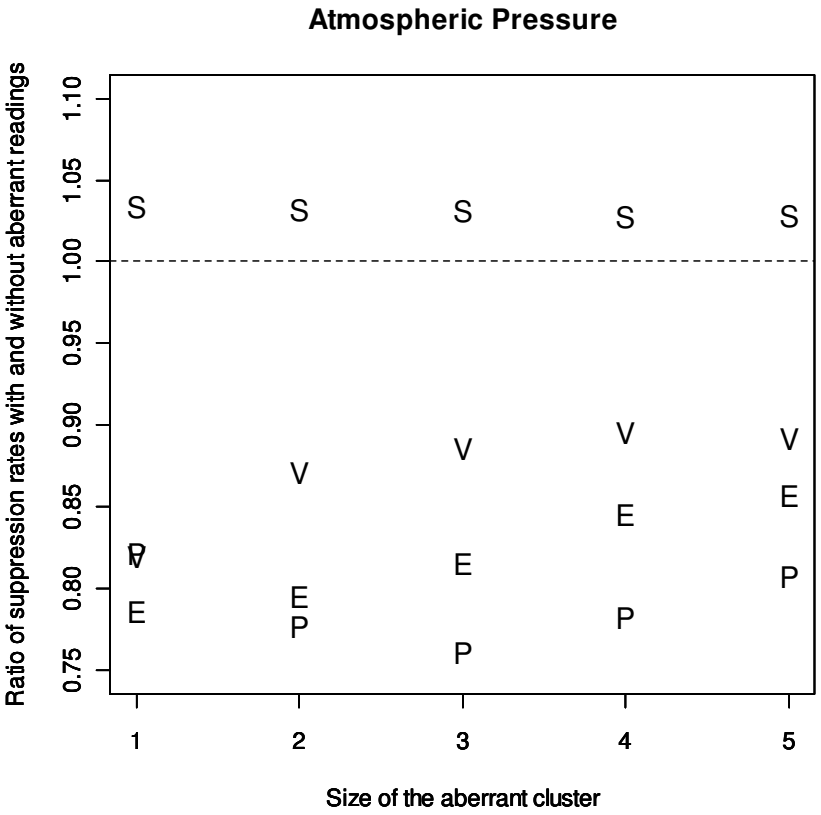


Figure 4.15 - Influence of aberrant readings on the suppression rate of the evaluated schemes applied on atmospheric pressure time series. Legend: S for TS-SOUND (r=0.1, T= 4, α =0.15), V for value-based, P for PAQ and E for EXP (A_{PAQ}=15).

For VB scheme, aberrant clusters make nodes send data to the base station at least two times: in the beginning and in the end of the cluster. Inside the cluster, aberrant readings tend to be similar to each other, which reduce data sending. This could explain why the influence of aberrant readings on the suppression rates has been smaller for aberrant clusters than for isolated aberrant readings. Clusters of aberrant readings would tend to amortize the initial and final data sending.

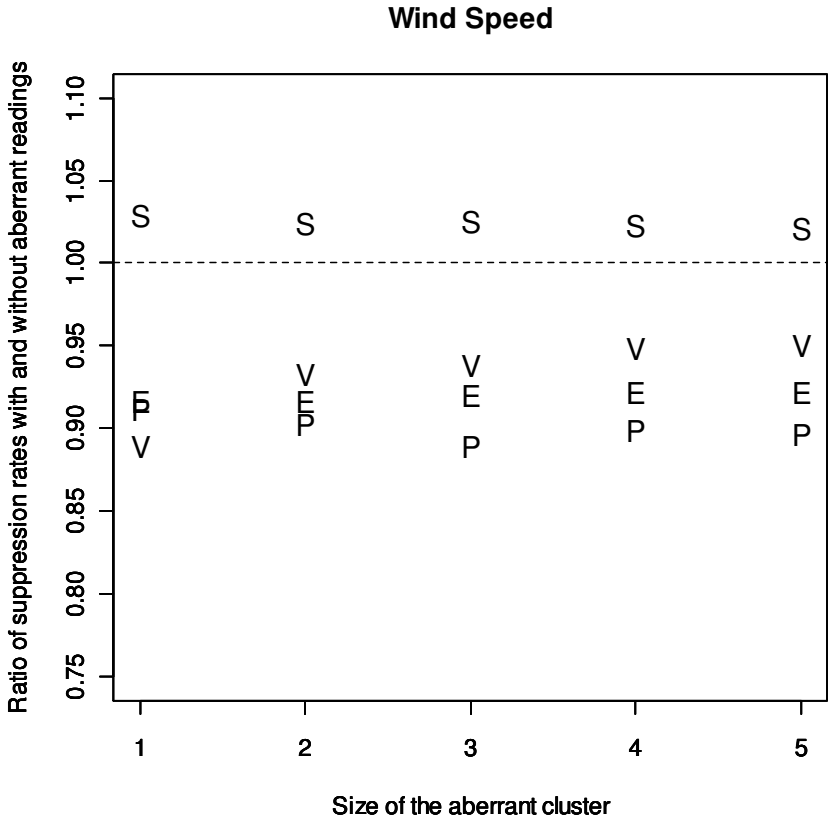


Figure 4.16 - Influence of aberrant readings on the suppression rate of the evaluated schemes applied on wind speed time series. Legend: S for TS-SOUND (r=0.1, T = 4, α =0.15), V for value-based, P for PAQ and E for EXP (A_{PAQ}=15).

4.7.5 A note on the order of the AR model

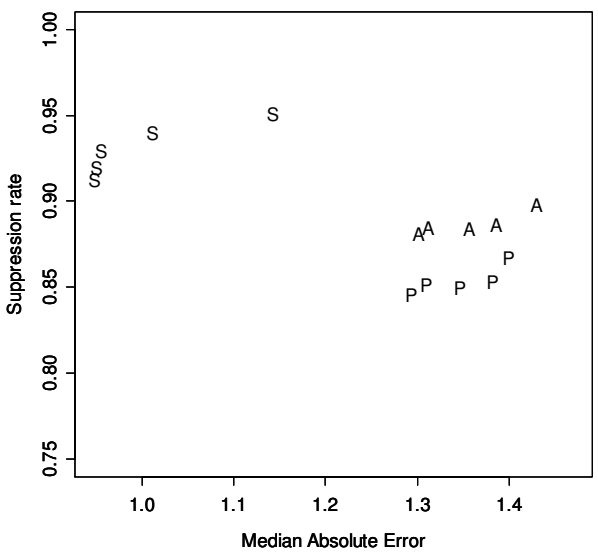
The model-driven approach is an efficient solution to data collection in sensor networks if the monitored variable has a well-known behavior so reliable models can be defined (SILBERSTEIN *et al.*, 2007a). Then, let us suppose that a sophisticated model is the best representation for the expected behavior of the sensor data. In this case, the simplicity of AR(1) model in the TS-SOUND scheme could degrade its performance if we compare it to the performance of a scheme adopting a more sophisticated model.

To evaluate this hypothesis, we have simulated time series according to the AR(3) model, which is the model that PAQ scheme uses. To generate the model coefficients, we have fit an AR(3) model to the time series in Figure 4.5. The series in Figure 4.5 represents the typical time series for each variable we have considered in the experiments. For each set of coefficients, we have simulated 50 time series with 1440 observations each, which corresponds to 50 days of monitoring with one reading per minute).

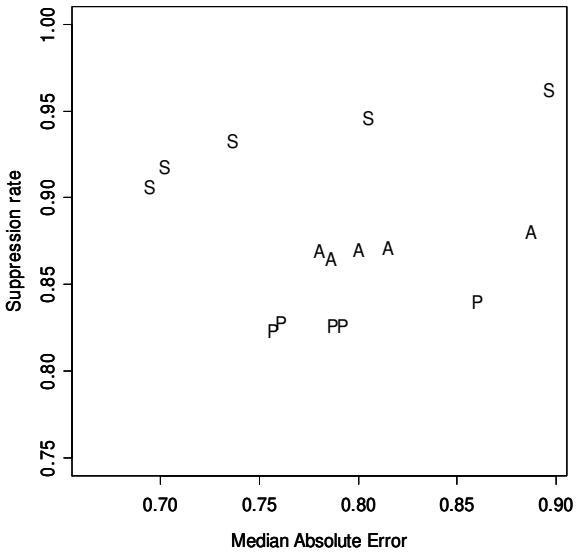
The simulated time series have presented different behaviors because the AR(3) coefficients used in the simulations have come from series with different behaviors (Figure 4.5). Since it is necessary to analyze the schemes' performances in groups of series with similar behaviors, we have had to quantify the differences between the behaviors of the simulated time series. To do this, we have defined the *Relative Lagged Difference* (RLD_l) as

$$RLD_l = \frac{\underset{t=l+1,l+2,...,N}{median}\left(\left|X_t - X_{t-l}\right|\right)}{\underset{t=1,..,N}{max}(X_t) - \underset{t=1,..,N}{min}(X_t)}, \quad l = 1,2,...,N-1 \quad . \tag{4.27}$$

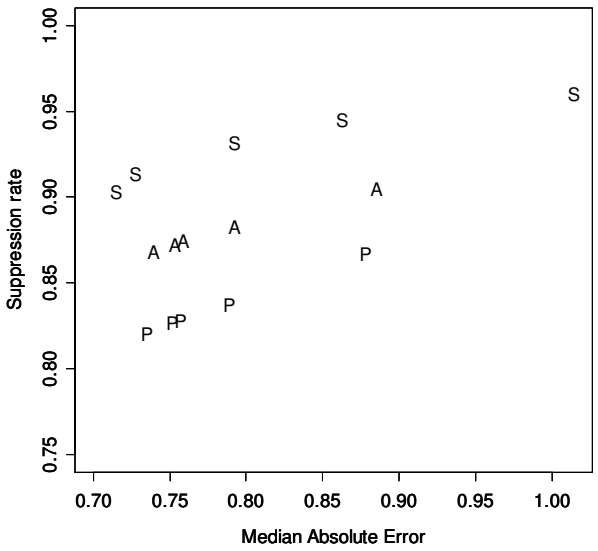
It compares the typical (median) difference between time periods t and t-l with the total range of the values. The values of RLD_l range from 0 to 1. The lag l indicates how local is the movement we want to capture. Smaller the value of l, the more localized the analysis. For instance, the values of RLD_{10} for the time series in Figure 4.5 are: 0.0942 (wind speed), 0.0252 (air temperature), 0.0201 (air relative humidity) and 0.0081 (atmospheric pressure). Therefore, time series with smooth changes relative to the total range (e.g., atmospheric pressure) have low values for RLD_l, whereas abrupt changes result in a higher value for RLD_l (e.g., wind speed).



(A) $0 \leq RLD_{10} < 0.025$



(B) $0.025 \leqslant RLD_{10} < 0.050$



(C) $0.05 \leqslant RLD_{10} < 0.075$

Figure 4.17 - Summaries for the performance of TS-SOUND and PAQ schemes in data simulated according to the AR(3) model. Legend: S for TS-SOUND (r=0.1, T=4), P and A for PAQ with A_{PAQ} = 5 and 15, respectively. Each point of a scheme represents the summary of its performance using a different value for α (0.15, 0.10, 0.05, 0.025, 0.01), in this order, following the increasing of the suppression rates. For PAQ scheme, the values for the correspondent error thresholds, ε_δ , have been defined as described in section 4.6.2.

After calculating the RLD_{10} for all 200 time series, we have separated them into three groups according to their RLD_{10} value and applied TS-SOUND and PAQ schemes on the time series of each group. The values for the parameters have been the same of the experiments in section 4.7.4.

Figure 4.17 presents the summaries for the performance of both schemes in the three groups of time series. Similarly to the figures of section 4.7.1, points closer to the upper-left corner represent the schemes with the best performances. As in the experiments with real data, PAQ scheme using the largest post-monitoring window (A_{PAQ}=15) have outperformed the schemes using a smaller time window (A_{PAQ}=5).

We expected that PAQ scheme could get at least prediction errors smaller than the errors of TS-SOUND. However, even in a scenario clearly favorable to PAQ, the most of TS-SOUND schemes have outperformed their correspondent PAQ schemes. In the time series with smooth changes relative to the total range (Figure 4.17A), all TS-SOUND schemes have outperformed all PAQ schemes, getting the highest suppression rates and the smallest prediction errors. As the time series have increased their local variation relative to their total range (RLD_{10} increases), PAQ schemes have got prediction errors closer to the errors of TS-SOUND schemes. However, for the first two values of α, TS-SOUND has still outperformed PAQ.

4.8 Discussion

Data suppression schemes are defined by an agreement between sensor nodes and base station about the expected behavior for the sensor readings. To decide when the sensor nodes may suppress their data, the schemes evaluate the prediction error, which is the difference between the value the sensor actually collects and the value predicted according to the expected behavior for the sensor readings. If the

collected value fits to the expected behavior, node suppresses its data. Otherwise, it sends data to the base station.

Since the schemes for data suppression look for changes in the expected behavior of the sensor data, they are sensitive to aberrant readings. Transmitting these erroneous data is a waste of energy. In a simple suppression scheme as the Value-based (SILBERSTEIN *et al.*, 2007a), for instance, an aberrant point may produce two unnecessary messages to the base station. That is because the scheme detects two sequential changes of behavior: one when the aberrant readings occur and another when the readings get normal again.

To avoid sending aberrant readings, one can propose to use a fixed threshold: readings smaller or greater a predefined value would be considered as erroneous data. However, that is a naive solution, since what would be aberrant at a time period of the series might not be aberrant at another time period. For instance, a reading of 1026 mb at time period 200 in the atmospheric pressure series (Figure 4.5) would be considered aberrant. However, this value should not be considered aberrant at time period 1000.

In this chapter, we have proposed TS-SOUND, a scheme for temporal data suppression in sensor networks that is robust to aberrant readings. TS-SOUND considers the data collected by a sensor node as a time series and monitors the behavior of this series. It adopts a procedure to detect outliers from a time series and the posterior classification of the detected outlier into a change-point or an aberrant reading. In the former case, data are sent to the base station, since it means a change in the expected behavior of the data series. Otherwise, data are suppressed.

Schemes for temporal data suppression proposed in sensor networks literature (PAQ (TULONE and MADDEN, 2006), EXP and Value-based (SILBERSTEIN *et al.*,

2007a)) suppress data by comparing the absolute value of the prediction error with a fixed threshold. Using the absolute value of the prediction error allows for controlling its maximum value. However, if the random fluctuations around the expected value (local variations) are larger than the threshold for the absolute error, a large amount of unnecessary data will be sent to the base station and the suppression rates will be small. On the other hand, if the local variations are smaller than the threshold for the absolute error, the suppression scheme will not be able to capture changes in the expected behavior of the monitored data. Then, if the time series has a nonstationary variance, a fixed threshold for the absolute prediction error will not be able to work well during all data collection.

TS-SOUND scheme also uses an error measure to decide if an observation is an outlier. However, it adopts a relative error measure, comparing the absolute error with the data variance, which captures the random fluctuations of the data. As a result, TS-SOUND is able to be adaptable to the local variations of the time series. The suppression rates of TS-SOUND scheme are more robust to the size of the local variations than the other schemes evaluated in this chapter.

Besides adopting the relative prediction error, TS-SOUND scheme tries to minimize its sensitivity to aberrant readings using the past data through a moving average. Moreover, even if an aberrant reading raises the outlier alarm, TS-SOUND opens a post-monitoring window to avoid sending this erroneous data to the base station. Although this post-monitoring window introduces a delay in the data delivery, our experiments have shown that a small delay (four time periods) can deal with time series presenting aberrant clusters of several sizes.

Using real data from several sources, which presents different temporal behaviors, we have run experiments to evaluate the suppression rates of TS-SOUND scheme and the prediction errors attached to them. We have used both of these measures to quantify the performance of a data suppression scheme. We have also

evaluated TS-SOUND's robustness to aberrant readings and compared its performance with the performance of PAQ, EXP and VB schemes. The evaluation experiments have shown that TS-SOUND is more robust to aberrant readings than the other schemes considered in this chapter. Moreover, TS-SOUND has outperformed the model-based suppression schemes (PAQ and EXP) in all evaluated scenarios and VB scheme in the most of these situations.

The Value-Based is the simplest suppression scheme and has got one of the best performances in our experiments. However, we can list at least three situations in which using TS-SOUND would be better than using Value-Based scheme: a) when the applications is not interested in aberrant readings; b) when the series presents different behaviors along the time, since VB uses a fixed error threshold and TS-SOUND is adaptable to the local variation of the time series; c) when having high suppression rates is more important than having small prediction errors.

To define a TS-SOUND suppression scheme, the user has to choose the values for three parameters: the weight of the last sensed data (r) in the on-line estimation of the distribution parameters, the length of the post-monitoring and past time windows (T) and the rigor to classify an observation as an outlier (α). As we have discussed in section 4.7, we have found that the value of T has not to be as large as the cluster size. Our experiments have pointed out to 4 as the smallest value for T that leads to homogeneous performances in time series with different behaviors and several sizes of aberrant clusters. On the value of r, our experiments have shown that we obtain the best performance for TS-SOUND when the on-line estimation of the new values for the distribution parameters sets less weight to the current sensor reading. TS-SOUND schemes using $r=0.1$ have produced the best results and values of r smaller than 0.1 have got very similar results. However, weights larger than 0.1 have degraded the suppression rates.

Since the values for T and r can be constrained to some predefined values, the network user has to choose only the value for α. To do this, it is necessary to define what is more crucial: capturing small changes (large values for α) or avoid aberrant readings (small values for α).

The main contributions of this chapter are two-fold: a proposal for a data suppression scheme that is robust to aberrant readings and the evaluation of the performance of data suppression schemes considering not only the saved energy but also the quality of the data collected at base station.

4.9 Future Directions

Sensor networks collect spatially correlated data, which produces areas in the sensors field that are spatially homogeneous. Our future work includes a spatio-temporal version of the TS-SOUND scheme having as its spatial basis the clustering algorithm in REIS *et al.* (2008). Instead of sending its reports to the base station, the nodes organize themselves into clusters that explore the spatial homogeneity of the data in the sensors field. Besides localizing the most part of the communication among the nodes, such clusters improve the quality of the cluster data summaries to be sent to the base station (REIS *et al.*, 2007).

The nodes of a sensor network are prone to failures as well as the communication between nodes can be very noisy. Thus, a data collection protocol based on a suppression scheme has to address an important question: how can we distinguish suppressed reports from nodes failures and lack of communication between nodes and base station? Silberstein et al. (2007b) have proposed interesting alternatives to deal with this problem using Bayesian inference. We study to incorporate the proposed solutions in the spatio-temporal version of TS-SOUND scheme.

5 CONCLUSION

Sensor networks promise to revolutionize the collection of environmental data and are considered to be the "next step in the understanding of the environment" (HART and MARTINEZ, 2006). Working in self-contained applications or being part a heterogeneous network, as the NASA SensorWeb (CHIEN *et al.*, 2005), sensor networks represents a huge advance for the environmental data collection.

As an emerging technology, sensor networks pose many challenges for several science disciplines. One of them arises on the smart use of the limited energy of the sensor nodes. This issue is crucial to warrant the main goal of a sensor network: to deliver data with an acceptable quality while saving the nodes' energy to prolong the network's lifetime.

In this thesis, we have examined the data suppression as a strategy to collect data using a sensor network. Our goal has been to improve the quality of data estimates delivered at the network's base station using a data suppression scheme. Besides, we have proposed to use the statistical quality of the estimates as an additional metric to evaluate the performance of a data collection proposal. Therefore, a trade-off between the statistical quality of the estimates and the energy consumption should be used in the proposal's evaluation.

We have investigated two strategies to suppress data: spatial suppression (cluster-and-aggregate) and temporal suppression.

In Chapter 2, we have found that spatially homogeneous clusters produce averages that estimate the members' data better than non data-aware clusters. In that chapter, we have used a centralized clustering procedure.

Based on the conclusions of Chapter 2 and considering the distributed feature of a sensor network, Chapter 3 has presented our proposals DARC

and DA-DCA. They are two distributed clustering algorithms that improve the quality of the data estimates if compared with usual distributed data-aware clustering procedures.

In Chapter 4, we have presented TS-SOUND, which is a temporal suppression scheme to deal with the occurrences of aberrant readings. By filtering these erroneous data based on the expected behavior of the data series, TS-SOUND has got a trade-off between suppression rates (energy saving) and prediction error (quality of the estimates) that is comparable and even superior to the trade-off of other proposal for temporal data suppression. Besides, we have shown that TS-SOUND is more robust to aberrant readings.

As a future work, we are preparing a proposal for spatio-temporal data suppression scheme putting together the proposals presented in chapters 3 and 4.

In this thesis, we have worked with static sensor networks. Our future work also includes adapting the presented proposals to networks composed by mobile sensor nodes.

REFERENCES

AKKAYA, K.; YOUNIS, M. A survey on routing protocols for wireless sensor networks. **Ad Hoc Networks**, v. 3, n. 3, p. 325 -- 349, 2004.

ASSUNÇÃO, R. M.; NEVES, M. C.; CÂMARA, G.; FREITAS, C. C. Efficient regionalisation techniques for socio-economic geographical units using minimum spanning trees. **International Journal of Geographical Information Science**, v. 20, n. 7, p. 797--811, 2006.

BASAGNI, S. Distributed Clustering for Ad Hoc Networks. In: International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN'99), 4., Fremantle, Australia. **Proceedings...** Fremantle: IEEE, 1999. p. 310--315. ISBN ISBN:0-7695-0231-8.

BIAGIONI, E.; BRIDGES, K. The application of remote sensor technology to assist the recovery of rare and endangered species. **International Journal of High Performance Computing Applications**, v. 16, n. 3, p. 315 -- 324, 2002.

BINS, L.; FONSECA, L.; ERTHAL, G. Satellite Imagery Segmentation: a region growing approach. In: VIII Brazilian Symposium on Remote Sensing, São José dos Campos, BR. **Proceedings...** INPE, 1996. p. 677 - 680.

BRANCH, J.; SZYMANSKI, B.; GIANNELLA, C.; WOLFF, R.; KARGUPTA, H. In-network outlier detection in wireless sensor networks. In: International Conference on Distributed Company Systems (ICDCS), 26. , July 4-7, 2006. Lisboa, Portugal. **Proceedings...** Lisboa: IEEE, 2006.

CAMARA, G.; SOUZA, R. C. M.; FREITAS, U. M.; GARRIDO, J. SPRING: Integrating remote sensing and GIS by object-oriented data modelling. **Computers & Graphics**, v. 20, n. 3, p. 395 -- 403, 1996.

CARDELL-OLIVER, R.; SMETTEMY, K.; KRANZZ, M.; MAYERX, K. A Reactive Soil Moisture Sensor Network: Design and Field Evaluation. **International Journal of Distributed Sensor Networks**, v. 1, n. 2, p. 149 -- 162, 2005.

CHIEN, S.; CICHY, B.; DAVIES, A.; TRAN, D.; RABIDEAU, G.; CASTANO, R.; SHERWOOD, R.; MANDL, D.; FRYE, S.; SHULMAN, S.; JONES, J.; GROSVENOR, S. An Autonomous Earth-Observing Sensorweb. **IEEE Intelligent Systems**, v. 20, n. 3, p. 16 -- 24, 2005.

CHU, D.; DESHPANDE, A.; HELLERSTEIN, J. M.; HONG, W. Approximate Data Collection in Sensor Networks using Probabilistic Models. In: International Conference on Data Engineering (ICDE'06), 22., Apr 3 - 8, 2006. Atlanta, GA. **Proceedings...** Atlanta: IEEE, 2006. p. 129 -- 136.

CULLER, D.; ESTRIN, D.; SRIVASTAVA, M. Overview of Sensor Networks. **IEEE Computer**, v. 37, n. 8, p. 41 - 49, 2004.

DRAPER, N. R.; SMITH, H. **Applied linear regression.** 3 ed. New York: John Wiley & Sons, 1998. 700 p.

ELSON, J.; ESTRIN, D. Sensor networks : a bridge to the physical world. In: RAGHAVENDRA, C. S.; SIVALINGAM, K. M.; ZNATI, T. (Ed.). **Wireless Sensor Networks**. New York: Kluwer, 2004.

ENVISENSE-SECOAS. **Self-organizing Collegiate Sensor Networks**. Disponível em: http://envisense.org/secoas.htm. Acesso em: 05/12/2005.

FRERY, A. C.; ALENCARNETO, J.; NAKAMURA, E. Error Estimation in Wireless Sensor Networks. In: ACM Symposium on Applied Computing (ACM SAC2008), 23., Fortaleza (CE), Brazil. **Proceedings...** Fortaleza: ACM, 2008. p. 1923 -- 1928.

FRISÉN, M. Statistical Surveillance. Optimality and Methods. **International Statistical Review** v. 71, n. 2, p. 403–434, 2003.

GOLDIN, D. Faster In-Network Evaluation of Spatial Aggregation in Sensor Networks. In: International IEEE Conference On Data Engineering (ICDE'06), 22., Apr 3 - 8, 2006. Atlanta, GA. **Proceedings...** Atlanta: IEEE, 2006. p. 148 -- 148.

GRUBBS, F. E. Procedures for Detecting Outlying Observations in Samples. **Technometrics**, v. 11, n. 1, p. 1 -- 21, 1969.

HART, J. K.; MARTINEZ, K. Environmental Sensor Networks: A revolution in the earth system science? **Earth-Science Reviews**, v. 78, n. 3-4, p. 177–191, 2006.

HEINZELMAN, W. B.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Hawaii International Conference on System Science, 33., Hawaii, USA. **Proceedings...** Los Alamitos: IEEE, 2000. p. 1--10. CD-ROM.

___. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. **IEEE Transactions On Wireless Communications**, v. 1, n. 4, p. 660--670, 2002.

HODGE, V.; AUSTIN, J. A Survey of Outlier Detection Methodologies. **Artificial Intelligence Review**, v. 22, n. 2, p. 85--126, 2004.

IBRIQ, J.; MAHGOUB, I. Cluster-based routing in wireless sensor networks: issues and challenges. In: Symposium on Performance Evaluation of Computer

Telecommunication Systems, San Jose, CA. **Proceedings...** 2004. Disponível em: http://www.scs.org/scsarchive/search.cfm?presearch=db&dbrec=40.

JUANG, P.; OKI, H.; WANG, Y.; MARTONOSI, M.; PEH, L.; RUBENSTEIN, D. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In: International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), 10., San Jose (CA), USA. **Proceedings...** San Jose: ACM, 2002. p. 96 -- 107.

KOTIDIS, Y. Snapshot Queries: Towards Data-Centric Sensor Networks. In: International Conference on Data Engineering (ICDE'05), 21., Tokyo, Japan. **Proceedings...** Tokyo: IEEE, 2005. p. 131 -- 142.

KOTIDIS, Y.; DELIGIANNAKIS, A.; STOUMPOS, V.; VASSALOS, V.; DELIS, A. Robust Management of Outliers in Sensor Network Aggregate Queries. In: International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'07), 6., Beijing, China. **Proceedings...** Beijing: ACM, 2007. p. 17 -- 24.

LEHMAN, E. L. **Testing statistical hypothesis.** Springer Texts in Statistics. 2 ed. Berlin: Springer, 1997. 769 p.

LINDSEY, S.; RAGHAVENDRA, C. S. PEGASIS: Power Efficient GAthering in Sensor Information Systems. In: IEEE Aerospace Conference, Mar. 2002. Big Sky, EUA. **Proceedings...** Big Sky: IEEE, 3, 2002. p. 1125 -- 1130.

MADDEN, S.; FRANKLIN, M. J.; HELLERSTEIN, J. M.; HONG, W. Tag: a tiny aggregation service for ad-hoc sensor networks. **ACM SIGOPS Operating Systems Review**, v. 36, n. SI, p. 131 -- 146, 2002.

MAINWARING, A.; POLASTRE, J.; SZEWCZYK, R.; CULLER, D.; ANDERSON, J. Wireless Sensor Networks for Habitat Monitoring. In: ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, EUA. **Proceedings...** Atlanta: ACM, 2002. p. 88 – 97.

MANJESHWAR, A.; AGRAWAL, D. P. TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks. In: International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, 1., Apr. 2001. San Francisco (CA), EUA. **Proceedings...** 2001.

MARTINEZ, K.; HART, J. K.; ONG, R. Environmental Sensor Networks. **IEEE Computer**, v. 37, n. 8, p. 50 - 56, 2004.

MCGILL, R.; TUKEY, J. W.; LARSEN, W. A. Variations of Boxplots. **The American Statistician**, v. 32, n. , p. 12--16, 1978.

MUTHUKRISHNAN, S.; SHAH, R.; VITTER, J. S. Mining Deviants in Time Series Data Streams. In: International Conference on Scientific and Statistical Database Management (SSDBM '04), 16., Santorini Island, Greece. **Proceedings...** Greece: IEEE, 2004. p. 41.

NAKAMURA, E. F.; LOUREIRO, A. A. F.; FRERY, A. C. Information Fusion for Wireless Sensor Networks:Methods, Models, and Classifications. **ACM Computing Surveys**, v. 39, n. 3, p. 1 -- 55, 2007.

NITTEL, S.; STEFANIDIS, A. GeoSensor Networks and Virtual GeoReality. In: NITTEL, S., STEFANIDIS, A. (Ed.). **GeoSensors Networks**. CRC Press, 2005, p. 296.

PADHY, P.; MARTINEZ, K.; RIDDOCH, A.; ONG, H. L. R.; HART, J. K. Glacial Environment Monitoring using Sensor Networks. In: Real-World Wireless Sensor Networks, Stockholm, Sweden. **Proceedings...** 2005. Disponível em: http://www.sics.se/realwsn05/papers/martinez05glacial.pdf.

PALPANAS, T.; PAPADOPOULOS, D.; KALOGERAKI, V.; GUNOPULOS, D. Distributed deviation detection in sensor network. **ACM SIGMOD Record**, v. 32, n. 4, p. 77-- 82, 2003.

POLLAK, M.; SIEGMUND, D. Sequential detection of a change in a normal mean when the initial value is unknown. **The Annals of Statistics**, v. 19, n. 1, p. 394--416, 1991.

POTTIE, J.; KAISER, W. J. Embedding the internet wireless integrated network sensors. **Communications of the ACM**, v. 43, n. 5, p. 51 – 58, 2000.

R_DEVELOPMENT_CORE_TEAM. R: A language and environment for statistical computing. Vienna, Austria.: R Foundation for Statistical Computing, 2005.

RAMASWAMY, S.; RASTOGI, R.; SHIM, K. Efficient Algorithms for Mining Outliers from Large Data Sets. **ACM SIGMOD Record**, v. 29, n. 2, p. 427 -- 438, 2000.

REIS, I. A.; CÂMARA, G.; ASSUNÇÃO, R. M.; MONTEIRO, A. Distributed Data-Aware Representative Clustering for Geosensor Networks Data Collection. In: Brazilian Workshop on Real-Time and Embedded Systems (WRT 2008), 10., Rio de Janeiro, RJ, Brazil. **Proceedings...** Rio de Janeiro: SBC, 2008. p. 77 -- 84. 1 CD-ROM.

REIS, I. A.; CÂMARA, G.; ASSUNÇÃO, R. M.; MONTEIRO, A. M. V. Data-Aware Clustering for Geosensor Networks Data Collection. In: Brazilian Remote Sensing Symposium, 13., 21-26 Apr. 2007. Florianópolis (SC), Brazil. **Proceedings...** São José dos Campos: INPE, 2007. p. 6059--6066. Disponível

em: http://marte.dpi.inpe.br/col/dpi.inpe.br/sbsr@80/2006/11.15.23.33/doc/6059-6066.pdf.

RYEL, R. J.; CALDWELL, M. M.; LEFFLER, A. J.; YODER, C. K. Rapid Soil Moisture Recharge to Depth by Roots in a Stand of *Artemisia Tridentata*. **Ecology**, v. 84, n. 3, p. 757–764, 2003.

SANTORO, N. **Design and analysis of distributed algorithms.** Wiley series on parallel and distributed computing. 1 ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2007. 583 p.

SCHLATHER, M., 2001, **Simulation of stationary and isotropic random fields**, R News, p. 18--20.

SILBERSTEIN, A.; BRAYNARD, R.; FILPUS, G.; PUGGIONI, G.; GELFAND, A.; MUNAGALA, K.; YANG, J. DataDriven Processing in Sensor Networks. In: Biennial Conference on Innovative Data Systems Research, 3., Asilomar (CA), USA. **Proceedings...** 2007a. p. 10--21.

SILBERSTEIN, A.; PUGGIONI, G.; GELFAND, A.; MUNAGALA, K.; YANG, J. Suppression and Failures in Sensor Networks: A Bayesian Approach. In: Very Large Data Bases (VLDB '07), 23-28 Sep. 2007. Vienna, Austria. **Proceedings...** Vienna: VLDB Endowment, 2007b. p. 842 -- 853

SINGH, M. P.; GORE, M. M. A New Energy-efficient Clustering Protocol for Wireless Sensor Networks. In: International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP2005), 2., 5-8 Dec. 2005. Melbourne, Australia. **Proceedings...** Melbourne: IEEE, 2005. p. 25-- 30.

SUBRAMANIAM, S.; PALPANAS, T.; PAPADOPOULOS, D.; KALOGERAKI, V.; GUNOPULOS, D. Online Outlier Detection in Sensor Data Using NonParametric Models. In: 2nd International Conference on Very Large Data Bases (VLDB'06), September 12--15, 2006. Seoul, Korea. **Proceedings...** Seoul: VLDB Endowment, 2006. p. 187 -- 198.

SUBRAMANIAN, L.; KATZ, R. H. An Architecture for Building Self Configurable Systems. In: IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA. **Proceedings...** Boston: IEEE, 2000. p. 63 -- 73. ISBN 0-7803-6534-8.

SZEWCZYK, R.; POLASTRE, J.; MAINWARING, A.; CULLER, D. Lessons From A Sensor Network Expedition. In: First European Workshop on Sensor Networks (EWSN), Berlin, Germany. **Proceedings...** 2004. p. 307 -- 322.

TATESON, J.; ROADKNIGHT, C.; GONZALEZ, A.; KHAN, T.; FITZ, S.; HENNING, I.; BOYD, N.; VINCENT, C. Real World Issues in Deploying a Wireless Sensor Network for Oceanography. In: Workshop on Real-World Wireless Sensor

Networks (REALWSN'05), Stockholm, Sweden **Proceedings...** 2005. Disponível em: http://www.sics.se/realwsn05/.

TENNEY, R. R.; SANDELL JR., N. R. Detection with distributed sensors. **IEEE Transactions on Aerospace and Electronic Systems**, v. 17, n. 4, p. 501–510, 1981.

TILAK, S.; ABU-GHAZALEH, N. B.; HEINZELMAN, W. A taxonomy of wireless micro-sensor network models. In: ACM Workshop on Wireless Security, Atlanta, USA. **Proceedings...** Atlanta: ACM, 2002. p. 28 -- 36.

TOLLE, G.; POLASTRE, J.; SZEWCZYK, R.; CULLER, D.; TURNER, N.; TU, K.; BURGESS, S.; DAWSON, T.; BUONADONNA, P.; GAY, D.; HONG, W. A Macroscope in the Redwoods. In: ACM Conference on Embedded Networked Sensor Systems (SenSys'05), 3., November 2–4, 2005. San Diego (CA), USA. **Proceedings...** San Diego: ACM, 2005. p. 51 -- 63.

TUKEY, J. W. **Exploratory data analysis.** 1 ed. Reading, MA: Addison-Wesley, 1977. 688 p.

TULONE, D.; MADDEN, S. PAQ: Time Series Forecasting For Approximate Query Answering In Sensor Networks. **Lecture Notes in Computer Science**, v. 3868, n., p. 21--37, 2006.

WALD, L. Some terms of reference in data fusion. **IEEE Transactions on Geoscience and Remote Sensing**, v. 13, n. 3, p. 1190 -- 1193, 1999.

WARNEKE, B.; LAST, M.; LIEBOWITZ, B.; PISTER, K. S. J. Smart Dust: Communicating with a Cubic-Millimeter Computer. **Computer**, v. 34, n. 1, p. 44 – 51, 2001.

WERNER-ALLEN, G.; LORINCZ, K.; WELSH, M.; MARCILLO, O.; JOHNSON, J.; RUIZ, M.; LEES, J. Deploying a Wireless Sensor Network on an Active Volcano. **IEEE Internet Computing**, v. 10, n. 2, p. 18 -- 25, 2006.

XU, N. A Survey of Sensor Network Applications. **IEEE Communications Magazine**, v. 40, n. 8, p. 102 – 114, 2002.

YAMANISHI, K.; TAKEUCHI, J.-I. A Unifying Framework for Detecting Outliers and Change Points from Non-Stationary Time Series Data. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 8., Alberta, Canada. **Proceedings...** Alberta: ACM, 2002. p. 676 -- 681. ISBN 1-58113-567-X.

YOUNIS, M.; YOUSSEF, M.; ARISHA, K. Energy-Aware Routing in Cluster-Based Sensor Networks. In: IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems

(MASCOTS2002), 10., Oct. 2002. Fort Worth, TX. **Proceedings...** Fort Worth: IEEE, 2002. p. 129 -- 136.

APPENDIX - DARC ALGORITHM

Figure A.1 presents DARC algorithm using the methodology described in SANTORO (2007) to specify protocols and design distributed algorithms.

The notation is the same we adopted in section 3.2 (figures 3.1 and 3.2). However, it is worth to mention that:

1. a sensor node is denoted by lowercase letters (e.g. x, y or j);
2. c(x) is the alarm clock of node x;
3. v(x) is the value sensed by node x;
4. ID(x) is the ID of node x;
5. N(x) is list of the geographical neighbors of node x and N'(x) is the list of the similar neighbors of node x;
6. MAD(x) is the mean absolute deviation of the values sensed by node c.

We thank to Sr. Marcelo Gabriel Almiron (Universidade Federal de Alagoas – UFAL, Brazil) to make this description available.

```
        PROTOCOL    DARC

1       Status values:  S = {HEAD,NON-HEAD,SOLITARY,INITIATOR}
2                       S_init = {INITIATOR}
3                       S_term = {HEAD,NON-HEAD,SOLITARY}
4       Restrictions:    SYNCRONIZED CLOCKS, BIDIRECTIONAL LINKS
5
6
7       INITIATOR
8       When ( c(x) = alarm_initiator )
9       Begin
10        send <hello,ID(x),c=v(x)> to N(x);
11        become WAITING_NEIGHBORS_CANDIDATES;
12      End
13
14      Receiving <hello,ID(y),c=v(y)>
15      Begin
16        if ( | v(x)-v(y) | ≤ ( θ x MAD(x) ) ) then N'(x):= N'(x) ∪ ID(y);
17        become WAITING_NEIGHBORS_CANDIDATES;
18      End
19
20
21      WAITING_NEIGHBORS_CANDIDATES
22      Receiving <hello,ID(y),c=v(y)>
23      Begin
24        if ( |v(x)-v(y)| ≤ ( θ x MAD(x) ) ) then N'(x):= N'(x) ∪ ID(y);
25      End
```

Figure A.1 - Description of DARC as a distributed algorithm (*to be continued*).

```
26
27      When ( c(x) = alarm_initiator + Δ_TN )
28      Begin
29        id := nearest_neighbors( N'(x) );
30        send <near,c=id> to N(x);
31      CH_list(x) := CH_list ∪ id;
32      become WAITING_HEAD_CANDIDATES;
33      End
34
35      Receiving <near,c=id>
36      Begin
37        if ( id ∈ N'(x) ) then CH_list(x) := CH_list ∪ id;
38        become WAITING_HEAD_CANDIDATES;
39      End
40
41
42      WAITING_HEAD_CANDIDATES
43      Receiving <near,c=id>
44      Begin
45        if ( id ∈ N'(x) ) then CH_list(x) := CH_list ∪ id;
46      End
47
48      When ( c(x) = alarm_initiator + Δ_TT )
49      Begin
50        become SELECTING_HEAD;
51      End
52
```

Figure A.1 - Continuation (to be continued).

```
53
54      SELECTING_HEAD
55      Spontaneously
56      Begin
57        CH_list(x) := frenquently( CH_list(x) );
58        if ( CH_list(x) = empty ) then
59            CH_list(x) := N'(x);
60        if ( ID(x) ∈ CH_list(x) ) then
61            CH(x) := ID(x);
62            send <head,c=ID(x)> to N(x);
63            CH_count(x) := 1;
64            become HEAD;
65        if ( ID(x) ∉ CH_list(x) ) then become NON_HEAD;
66      End
67
68
69      NON_HEAD
70      Receiving <head,c=ID(y)>
71      Begin
72        if ( ID(y) ∈ CH_list(x) ) then CH_cand(x) := CH_cand(x) ∪ ID(y);
73        if ( ID(y) ∉ CH_list(x) AND ID(y) ∈ N'(x) ) then
74            CH_wait(x) := CH_wait(x) ∪ ID(y);
75      End
76
77      When ( c(x) = alarm_initiator + Δ_TT + Δ_TA )
78      Begin
79        CH(x) := nearest_member( CH_cand(x) );
```

Figure A.1 - Continuation (to be continued).

```
80        if ( CH_cand(x) = empty ) then
81            CH(x) := nearest_member( CH_wait(x) );
82            if ( CH_wait(x) = empty ) then
83                CH(x) := ID(x);
84                become SOLITARY;
85        if ( CH(x) ≠ ID(x) ) then send <join,CH(x),ID(x)> to CH(x);
86      End
87
88
89      HEAD
90      Receiving <join,ID(x),ID(y)>
91      Begin
92        if ( ID(y) ∈ N'(x) ) then
93            send <data,ID(x),ID(y),c=∅> to ID(y);
94            CL(x) := CL(x) ∪ ID(y);
95      End
96
97      Receiving <head,c=ID(x)>
98      Begin
99        CH_wait(x) := CH_wait(x) ∪ ID(y);
100     End
101
102     When ( c(x) = alarm_initiator + Δ_TT + Δ_TA + Δ_TJ )
103     Begin
104       if ( CL(x) = empty ) then
105           if ( CH_wait(x) = empty ) then
106                become SOLITARY;
107       Else
108           j = first( CH_wait(x) );
```

Figure A.1 - Continuation (*to be continued*).

```
109         send <join,j,ID(x)> to j;
110     End
111
112     When ( c(x) = alarm_initiator + Δ_TT + Δ_TA + Δ_TJ + Δ_TAD )
113     Begin
114       if ( CL(x) = empty ) then become SOLITARY;
115     End
```

Figure A.1 - Conclusion.

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.