Numéro d'ordre: 40659

THESIS

(Thèse de doctorat en cotutelle)

to obtain the
DOCTORAT de l'UNIVERSITE DE LILLE 1
and
DOCTORAT de L'UNIVERSITE LIBANAISE

Delivred by
L'Ecole Doctorale Sciences Pour l'Ingenieur- EDSPI (France)
and
L'Ecole Doctorale Sciences et Technologies- EDST (Lebanon)

Specialty: *Computer Science*

Presented and defended by *Alia GHADDAR*
on 02/12/2011

Title

Improving the Quality of Aggregation using data analysis in WSNs

Supervisors: Pr. Isabelle SIMPLOT-RYL
Pr. Abbas HIJAZI
Dr. Tahiry Razafindralambo
Dr. Samar Tawbi

Reviewers:	
Mrs. Francesca GUERRIERO	Professor - University of Calabria, Italy
Mr. Amiya NAYAK	Professor - University of Ottawa, Canada
Jury:	
Mrs. Francesca GUERRIERO	Professor - University of Calabria, Italy
Mr. Amiya NAYAK	Professor - University of Ottawa, Canada
Mr. Srdjan KRCO	Senior Research and Innovation Consultant - Ericsson, Serbia
Mrs. Isabelle SIMPLOT-RYL	Professor - University of Lille 1, INRIA, France
Mr. Tahiry RAZAFINDRALAMBO	Junior Researcher, INRIA, France
Mr. Abass HIJAZI	Professor - Lebanese University, Lebanon
Mrs. Samar TAWBI	Doctor - Lebanese University, Lebanon

Acknowledgment

This thesis arose out of three years of research in POPS group of Lille-1 University and the Doctorate School of Science and Technology of the Lebanese University. During this time, I have worked with many people whose contribution in many ways deserves special mention.

In the first place I would like to convey my gratitude to my two directors, from Lille University, Pr. Isabelle Simplot-Ryl for proposing the subject of the thesis as well as her supervision, advice, and guidance from the very early stage of this research. I also express gratitude to Pr. Abbas Hijazi, from the Lebanese University, for his encouragement and support from the initial to the final level of my work.

I am also heartily thankful to my co-director, the person who enabled me to develop an understanding of the subject. Dr. Tahiry Razafindralambo had followed me in all the details of my work making possible the accomplishment of the thesis. I thank his tremendous effort, patience and cooperation with me. I thank as well Dr. Samar Tawbi. She provided me unflinching encouragement and support in various ways. I would like to give also my thanks to Pr. David Simplot-Ryl. He also provided insightful discussions and suggestions about the research.

I gratefully thank the members of my PhD committee. Using their precious times, in the midst of all their activity, to assist to my thesis defense is of great importance.

I offer my regards and blessings to all of those who supported me in any respect during the completion of this thesis. I also thank my colleagues, lab mates and administrative staffs for their support and for the enjoyable time.

Finally, words alone cannot express the thanks I owe to my family, for their love, encouragement and continuous support not only throughout this thesis but in all my life.

Abstract

Wireless Sensor Networks (WSNs) are becoming largely adopted in diverse sectors such as health care, home automation, industry process control, object tracking, etc. The promise and application domain of this field continue to grow due to the emergence of embedded, small and intelligent sensor devices in our everyday life. These devices are getting smarter with their capability to interact with the environment or other devices, to analyze data and to make decisions. They have made it possible not only gather data from the environment, but also to bridge the physical and virtual worlds, assist people in their activities, while achieving transparent integration of the wireless technology around us. Along with this promising glory for WSNs, there are however, several challenges facing their deployments and functionality, especially for battery-operated sensor networks. For these networks, the power consumption is the most important challenge. In fact, most of WSNs are composed of low-power, battery-operated sensor nodes that are expected to replace human activities in many critical places, such as disaster relief terrains, active volcanoes, battlefields, difficult terrain border lands, etc. This makes their battery replacement or recharging a non-trivial task. Hence, the main concern is how to deal with their limited energy-resources, especially that the performance of the network strongly depends on their lifetime. Therefore, energy saving becomes a serious and critical requirement that can be gained by properly managing the energy resources.

We are concerned with the most energy consuming part of these networks, that is the communication. We propose methods to reduce the cost of transmission in energy-constrained sensor nodes. Our work aims at reducing the data transmission rate and overhead. For this purpose, we observe the way data is collected and processed to save energy during transmission. Our work is build on three basic axis: data estimation, data similarity detection and abnormal behaviors detection. Nodes, in our work, have the ability to accurately predict data and detect abnormal behaviors. During data collection, the intermediate nodes can detect correlation among data and perform efficient data fusion, in order to send the minimum amount of representative values to the destination. These values are expected to assist the sink, so it can accurately re-construct the original data from the minimum amount it receives. In our work, we try to strike the balance between saving energy during transmission and the quality of data (sensed, transmitted or re-constructed).

Based on the experimentation we made on different data series, results show that, depending on data type, our prediction model can reduce the transmission rate about 70%. Nonetheless, the transmission overhead can be reduced to about 20% in most used data types, using our correlation detection algorithm. As for the quality of data (sensed, predicted or re-constructed), results show that the relative errors for prediction were between 2% and 8%. Furthermore, data can be accurately re-constructed at the sink. The relative error between the source and sink estimations were in $[-7 \times 10^{-7}, 10^{-6}]$. Finally, the performance of our outliers detection algorithm was reflected, in our experimentation, by its ability to detect accurately 78% up to 90% of slipped anomalous values.

Résumé

Les réseaux de capteurs sans fil (WSNs) sont largement adoptés dans divers secteurs comme la médecine, la domotique, le contrôle de processus industriels, la localisation des objets, etc. Les domaines d'application de ces réseaux continuent à croître grâce à l'émergence de capteurs de plus en plus petits et de plus en plus intelligents dans notre vie quotidienne. Ces dispositifs interagissent avec l'environnement ou d'autres périphériques, pour analyser les données et produire de l'information. En plus de créer de l'information, les réseaux de capteurs permettent de créer des ponts entre le monde physique et le monde virtuel. Les capteurs permettent, ainsi, une intégration transparente de la technologie virtuelle autour de nous. Il existe cependant plusieurs défis autour des réseaux de capteurs, notamment au niveau de l'autonomie énergétique des capteurs. La consommation énergétique de ces petits objets devient le principale verrou technologique empechant leur déploiement à grande échelle. En effet, la plupart des réseaux de capteurs sont composés des nœuds de faible puissance et fonctionnant sur batterie. Ils sont souvent utilisé dans des zones géographiques dangereuse et peu accessible, tels que les volcans actifs, les champs de bataille, ou après une catastrophe naturelle etc. Ces zones critiques rendent le remplacement ou la recharge des batteries de chaque capteur difficile voire impossible. Ainsi, le principal défi dans les réseaux de capteurs est la gestion de la consommation énergétique, pour permettre d'accroître sa durée de vie.

Nous sommes intéressés par la partie la plus consommatrice d'énergie dans les réseaux de capteurs: la communication ou l'envoi et la réception de données. Nous proposons des méthodes pour réduire les transmissions des nœuds en réduisant le volume de données a transmettre. Pour cela, nous observons la maniere dont les données corrélées pour réduire le nombre de transmissions. Notre travail s'articule autour de trois axes fondamentaux: la prédiction des données, la détection de similarité des données et la détection des comportements anormaux. Les solutions utilisant les séries temporelles que nous avons développées permettent de prédire efficacement les données et de détecter des comportements anormaux. Nous proposons aussi un mécanisme d'agrégation de données dans lequel les nœuds intermédiaires (entre la source et la destination des données) observent la corrélation entre les valeurs pour effectuer une fusion efficace de ces données, afin de minimiser la quantité de données envoyées. Dans notre travail, nous essayons de trouver l'équilibre entre l'économie d'énergie et qualité des données car toute prédiction ou fusion provoque des pertes.

Nous avons lancés des expérimentations sur différentes séries de données. Les résultats ont montré que notre solution de prédiction permet de réduire le taux de transmission jusqu'à 70% par rapport à une solution sans prédiction. Quant à la qualité des données, les résultats montrent que les erreurs relatives de prédiction sont compris entre 2% et 8% par rapport aux valeurs réelles. Ce qui montre que les données peuvent être reconstruites avec précision au niveau de la destination. L'erreur relative entre les estimations des nœuds source et du puits se situe entre $[-7 \times 10^{-7}, 10^{-6}]$. Enfin, notre algorithme de détection des anomalies permet de détecter jusqu'à 90% des valeurs anormales.

Contents

List of Figures

List of Tables

If men liked shopping, they'd call it research.
$\sim\sim\sim\sim$*Cythina Nelms*

That's the nature of research, you don't know what in hell you're doing.
$\sim\sim\sim\sim$*'Doc' Edgerton*

Chapter 1

Introduction

Summary

Wireless Sensor Networks (WSNs) have become a highly active research area due to their increasing potential impact on the quality of people's life and the health of the planet. They are used in many application domains for monitoring and tracking purposes. The significant shift in sensors characteristics (size, cost, limited power-source, intelligence capabilities, etc.) as well as their deployment in harsh environment, make it challenge for these sensors to remain functional for a long period of time. These concerns, particularly the power resource ones, pose considerable technical challenges in data processing, communication, and sensor management. In this chapter, we highlight these challenges and briefly describe the motivation and contribution of this work.

Contents

1.1 Preliminary

The development of smart environments have been identified as one of the most important challenges for the 21st century [248]. Designing systems with robust intelligence that are capable of perceiving, reasoning, learning, and interacting with their environment has been a pervasive goal in many research domains [52, 235]. Starting from the Internet itself, to smart robotics and smart homes, now the Internet of Things (IoT) steel the lights to become the new revolution of the Internet [235]. “Things” can be computers, sensors, people, refrigerators, vehicles, mobile phones, passports, luggage, etc. Having the capability to address and communicate with each other and verify their identities, all these “things” will be able to exchange information and, if necessary, make decisions. They can sense, communicate, interact, exchange data, information and knowledge. For example, the plants decide to water themselves if they were thirsty. The alarm decides to ring earlier if there is traffic or bad weather. A stolen object can interact with us to tell us where it is. Objects get intelligence thanks to the fact that they can communicate information about themselves and they can access information that has been aggregated by other things. The intelligence capabilities rely first and foremost on sensory data from the real world that are processed, analyzed and transmitted to other objects for further processing and decision-making.

The information needed by smart environments are mostly provided by Wireless Sensor Networks (WSNs), which are responsible for sensing, due to their ability to provide unobtrusive wire-free communication. Wireless sensor networks consist of small nodes with sensing, computation, and wireless communications capabilities [117]. They provide information gathering based on the collective efforts of hundreds or thousands of wireless sensor nodes. Along with the trend towards smart environments, where objects are embedded and fused around us, sensor nodes are intended to be physically small (such as smart dust of 1 cubic millimeter [127]). These nodes are equipped with one or more sensors, a short-range radio transceiver, a small micro-controller, and a power supply in the form of a battery. Sensors will be used to measure everything from acceleration and location to temperature, air pollution, and health conditions. Depending on the application, sensors can be deployed on the ground, in the air, under water, on bodies, in vehicles, and inside buildings. As each node is a data source, clearly, a lot of data is being collected. The multi-sensor in one node can capture a variety of environmental data, including, for example a mine temperature, humidity and gas concentration. By combining these precise measurements, sensor nodes can effectively prevent accidents and facilitate mine construction. The node can also send a series of commands to control the mine fans, to achieve real-time monitoring and changing the mine environment. Implanting devices inside of bodies will become commonplace. Three million pacemakers already implanted in people worldwide and 600,000 additional ones are implanted each year [215]. Such devices will be configured to allow invested parties (doctors, insurance companies, etc.) to access patient information, gather data and alerts via the Internet.

The analysis of the data streams to get information to analyze and take appropriate actions, is one of the design challenges for intelligent objects. Nonetheless, the rapid analysis of data provides early warnings of significant events, from impending heart attacks to climate change. Sensor data will be used to inform and modify human behavior on a personal level (“hey, you need more water, more exercise, more sleep”), as well as on a collective level ("tax policies need to be adjusted to discourage or encourage people's behavior such as vehicle road speed, etc). Important issues should be considered to provide sensors a sufficient degree of smartness and collaborative functionality. Several challenges should be taken into consideration: The networking aspects such as low resources in terms of computation and energy consumption in sensor nodes (Section 1.3), as well as the quality of data management (collection, processing and storage). These concerns are somewhat related. The higher the data quality, the better will be the outcomes (rapid events-trigger, patient symptoms detection, motion tracking, etc). "Better" here means cheaper and of good quality. We mean: "cheaper" in terms of sensors lifetime and resource constraints management in wireless sensor networks as we will see later in the following sections. While "good quality" is related to many aspects in WSNs (Quality of Information [95], Quality of Decision making (QoDm) [182], quality of routing (QoR) [32, 128], etc.) (see Chapter 2).

The tight energy budgets of battery-operated sensors enforce energy efficient designs of hardware components, network communication and applications algorithms. Challenges on energy consumption are related to nodes' sensing, communication and processing capabilities. Amongst these, it was reported that communication is the most power harvesting [116, 230, 271]. Thus, most approaches focus on the way nodes communicate and exchange data in order to lengthen their lifetime and improve their functionality. Two issues are taken into consideration: reducing communication to save energy but exchanging useful data to improve applications performance. In this thesis, we are concerned with how data sensed and aggregated by sensors can be processed to increase the performance of Wireless Sensor Networks (WSNs) in terms of energy efficiency and quality of information. In the following sections, we describe key concepts (data, information and knowledge), then we observe three domains of energy consumption in wireless WSNs (sensing, communication and processing). Our work tackles the impact of data collection and processing on the communication cost reduction (*i.e.* the transmission cost). We aim at reducing the number of transmissions between nodes (*i.e.* transmission rate) as well as the number of transmitted bits (*i.e.* transmission overhead) for the sake of energy efficiency of the sensor nodes. Then we move to present the motivation of our work and we close this chapter with a summary of our thesis outline.

1.2 Data, information and knowledge

The terms Data and Information are frequently used and some people made ongoing confusion between the two terms. Depending on the context, the meanings and use of these words differ. Many definition have been given [153, 208, 227]. Both data and information are types of knowledge. Data is the lowest level of knowledge while information is the second level. Depending on [153], information is useful data that has been processed in such a way as to increase the knowledge of the person who uses the data. From data to information to knowledge, there is a continuous knowledge discovery process. Information can be used as input for higher knowledge discovery level; And knowledge itself may be processed to generate decisions and new knowledge [191]. In wireless sensor networks, observations and recordings are done to obtain data (the lowest abstract or a raw input, referred to as *sensor reading*, *sensor value*, or *sample*), which when processed, analyzed or arranged makes meaningful output called information. The quality level of this information and its management affect the monitoring system performance. For example, in robot localization and navigation, data coming from both static nodes and mobile nodes can help to decide the location of a robot. The processing of this data guides a robot to make a decision to reach an event location or a disaster environment [219, 262]. At a minimum basis, a real-time data analysis for rapid response remains a research challenge. Data has to be sent on time and should be efficient enough, so sensors can have the necessary reactions to meet users and applications needs [16, 95, 182]. In WSNs, information on nodes can be erroneous and may be affected by malfunctioning or anomalous nodes. Due to the critical nature of many applications (such as disaster monitoring, war events recognition and tracking, health monitoring system, etc.), the integrity and accuracy of the sensed data are of high research and practical importance [41]. Actually, data collected by nodes need to be examined dynamically to detect any abnormal events, to report useful information and to avoid unnecessary communication cost. Thus, an efficient analysis and usage of data affects the network performance and its energy consumption. In what follows, we observe the factors that drain the source of energy on sensor nodes.

1.3 Energy consumption in WSNs

Typically, the architecture of a wireless sensor node consists of four main components: sensing, processing, transmission, and power units (Figure 1.1 [117]). The sensing subsystem includes one or more sensors (with associated analog-to-digital converters) for data acquisition. The processing subsystem includes a micro-controller and memory for local data processing. This unit allows sensor nodes to collaborate with each other to carry out the assigned sensing tasks. The transmission unit is for wireless data communication. One of the most important components of a sensor node is the power unit, which may be supported by a power-scavenging unit such as solar cells. Depending on the specific application, sensor nodes may also include additional components such as a location finding system to determine their position, a mobilizer to change their location or configuration (*e.g.*, antenna's orientation), a power generator [117]. The development of sensor networks over the years and their integration in many application domains, have been changing the node architecture, size and weight. Additional constraints on the

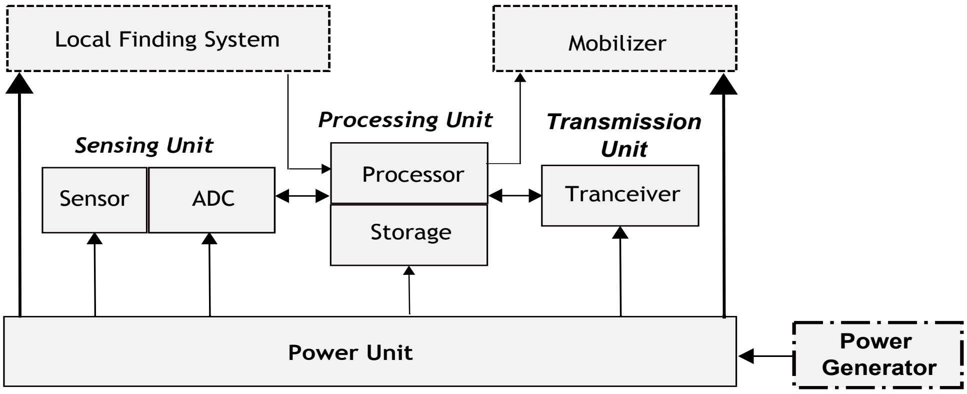


Figure 1.1: Sensor node architecture.

network performance have been imposed. The size of sensor node varies according to the application requirements. For example, the radar dome on AWACS planes (Airborne Warning and Control System), is 30 feet in diameter and six feet thick[1]. On the other hand, in health applications, a pacemaker is few cubic centimeters in size[2]). The current generation of wireless sensor hardware ranges from shoe-box sized such as Sensoria WINS NG sensors [154] with an SH-4 microprocessor to matchbox sized like Berkeley motes with an 8-bit microcontroller [116]. For smaller-size sensor nodes, the smart Dust team, at the university of california, demonstrated the possibility of a millimeter scale node called “smart dust” [127]. They also work on microflying insect or smart dust with legs[3]. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the complexity of the individual sensor nodes [189]. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth [189]. With the popularity of light-weight and small-size sensor nodes, challenges occur on their source of energy and lifetime. In what follows, we observe challenges related to energy-limited sensors. We detail the aspects that impact on the battery lifetime of sensors. These aspects are: sensing, communication and data processing.

1.3.1 Battery-operated sensors

Most of today’s sensors are battery operated. This imposes challenges on their functionality because batteries are the only limited life source available to power these sensors. Often, it is difficult even impossible to recharge batteries due to environmental conditions, such as war, disaster, in-body, volcano monitoring, etc. Therefore, sensors have to be extremely power efficient. In cases where sensors should be unobtrusive, they need to be lightweight with small form factor. The size and weight of sensors is predominantly determined by the size and weight of batteries. Batteries often constitute more than 50% of the device’s weight and volume. For example, the Mica2 [82] sensor node, has a weight of 18g. However, its two AA batteries weigh between 20 to 30g each. Hence, there is tradeoffs between lifetime requirements. The small form and low weight factors of a sensor node. Battery-operated sensors of small size have more restricted lifetime then large ones. In Body Sensor networks (BSN), sensors are battery operated and are required to last longer without any need of maintenance for about six months[4]. So frequent battery changes for multiple body sensors would likely hamper user’s acceptance. Energy consumption is one of the most important performance metrics for Wireless Sensor Networks, because it is directly related to the operational lifetime of the network [93]. Due to the great attention that power consumption gets, it is an important issue to study how to keep survivability of nodes for the sake of network functionally and lifetime. The main task of a sensor node in a sensor field is to detect events, perform quick local data processing, and transmit data. Energy consumption in a sensor node can hence be divided into three domains [7] (Figure 1.2): sensing, communication and data processing. We will discuss each of these aspects in the next subsections.

[1] http://www.airforce-technology.com/projects/e3awacs/

[2] http://texasheart.org/HIC/Topics/Proced/pacemake.cfm

[3] http://www-bsac.eees.berkeley.edu

[4] http://texasheart.org/HIC/Topics/Proced/pacemake.cfm

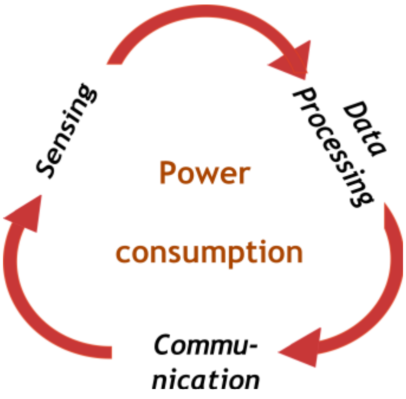


Figure 1.2: Sources of energy consumption in a sensor node.

1.3.2 Sensing

Some applications are sensing-constrained due to different factors [184] such as power hungry transducers, power hungry A/D converters, long-time data acquisition, etc. For example, Multimedia sensors [6] or biological sensors [71] are of power hungry transducers. They require more power to perform their data acquisition task (*i.e* sampling) [101]. Similarly, acoustic [209] and seismic transducers [250] require, generally, high-rate and high-resolution A/D converters. The power consumption of the converters can account for the most significant power consumption of the sensing subsystem, as in [199]. Energy consumption in data acquisition can attain 66% of the total energy consumption of a node, such as in H-mote with a Hybus sensor board that contains five set of air quality monitoring sensors [129]. Energy consumed in sensing unit can thus impact on the sensor performance (*i.e* its sensitivity and accuracy). Energy-efficient data acquisition techniques are used to reduce the energy consumption in sensing unit. Among these, we mention hierarchical sampling and adaptive sampling approaches [101]. Sampling is transforming the analog information of the object which is detected and controlled, to corresponding digital information. When a node is equipped with multiple sensors of different energy consumption, hierarchical sampling approaches are used. These techniques aim to dynamically select which sensor to activate, in order to get a trade off between accuracy and energy conservation. When an event is detected or a region has to be observed with greater detail, the accurate power-hungry sensors can be activated [176, 199].

The energy consumption of sampling is related to the sampling rate or frequency (i.e the number of samples per unit of time). Sampling with higher rate would generate more data and consumes more energy. Energy-efficient data acquisition techniques do not, exclusively, aim at reducing the energy consumption of the sensing subsystem. They decrease the number of communications as well, by reducing the data sampled by source nodes. Since samples can be correlated, adaptive sampling techniques can exploit this correlation to reduce the amount of data to be acquired from the transducer. For example, data of interest may change slowly with time, in this case, *temporal correlations* may be exploited to reduce the number of acquisitions [10]. A similar approach can be applied when the investigated phenomenon does not change sharply between areas covered by neighboring nodes. In this case, energy due to sampling (and communication) can be reduced by taking advantage from *spatial correlations* between sensed data [251].

In what follows, we consider that the sensing consumption is negligible, We tackle the data acquisition approaches related to communication rate reduction. More specifically, we are concerned with the data collection and aggregation techniques that reduce, as possible, the number of communication as well as the communicated overhead. In the next subsection, we observe the impact of the communication on the energy consumption of sensor nodes.

1.3.3 Communication

Typically, the operational states in a sensor node are: Transmit, Receive, Idle and Sleep [174, 211]. These states form a cycle called the duty-cycle. It is a periodic cycle between two states: wake-up state (transmit, receive, idle) and sleep state. Ideally, a node turns its radio ON only when it is going to transmit or receive data. At all other times, it should keep its radio off (*i.e* sleep state). When a radio is on but idle (waiting for possible traffic), it consumes the same amount of energy as when it is receiving data [183, 214]. The communication cost of a node is composed of the cost of transmission, reception and listening [155]. The cost of receiving or transmitting a packet is an order of magnitude greater than the cost of computation. Between the two (transmission and reception), transmitting a bit consumes twice as much energy as receiving a bit [116]. Tilak et al. show in [230] that the execution of 3000 instructions consumes the same amount of energy as sending 1 bit over 100 meters by radio. Therefore, the main concern for most WSNs algorithms [28, 146, 168, 256] is to keep the communication cost to a minimum. In this thesis, we focus on the transmission cost. We aim at reducing the number of transmissions between nodes (i.e the transmission rate) as well as the number of transmitted bits (i.e the transmission overhead).

In WSNs, data are transferred (i.e routed) from source sensors to the sink through multi-hop communication paradigm [7]. For an efficient routing, different schemes have been adopted, such as chain [137], tree [14] or clusters [264]. We aim at reducing the transmission cost on one-hop communication between nodes towards the base station. Description about these schemes is presented in Chapter 2. In this work, we suppose having a suitable routing protocols that do not interfere with the algorithms proposed in the following chapters.

1.3.4 Data processing

Nodes in WSNs gather, process and rely information to one or more other nodes. They collaborate to produce better outcomes during event monitoring. Ideally, data is expected to be accurate, arrive on time, processed and transmitted with lower energy consumption. The amount of data exchanged between nodes is expected to be as minimum as possible. However, data should be significant to respond to applications needs. For these purposes, collected data need to be processed and analyzed in a way to remove any erroneous, suspicious, redundant or other readings that may cause unnecessary computation and communication cost. Challenges do not stop here. They continue to grow along with the trends of producing mobile, smart or autonomous objects [235, 272]. Such objects require intelligent data processing algorithms to avoid bad decisions that affect their energy consumption. For example, an unreliable mobile node produces unpredictable motion and behaviors. It may decide to move in the wrong direction or take the longest route toward a destination. Bad decisions may harm its source of energy. Malicious nodes may divert traffic from the base station, drain energy of other nodes by producing fake information. They may modify the collected data to cause the base station to make mis-informed decisions. Such decisions are related to errors in data collection or errors in data processing, or shortly the quality of data collection and processing. The collection of undetected erroneous or incomplete data (due to data loss, misbehaviors, etc.) affects the quality of data analysis and the decisions made by sensors. Another negative impact is caused by the presence of data redundancy. Nodes that are densely deployed (20 nodes/m^3 [207]), and sense the same property are expected to provide the same information. The transmission of redundant data may waste energy. However, if data is filtered and redundancies are detected, the communication overhead can be reduced and energy consumption can be omitted. Fusion techniques were proposed to reduce the amount of data traffic, filter noisy measurements, and make predictions about a monitored entity [163, 164]. However, these techniques can lose much of the original structure in the data during compression, aggregation or prediction. Therefore, they need to be convenient with the required quality of data to prevent errors. For example, when sensor readings, from a number of sensors, are aggregated into one value by taking the average over all these readings. This value is used to represent each individual sensor reading. However, it deviates from the individual sensor readings and thus introduces an error. Similarly, a loss of data may occur during compression or predictions. In Chapter 2, we observe different data processing techniques for WSNs, such as, data aggregation, compression, prediction, anomaly detection. In the following section, we present the motivation and contribution of our work.

1.4 Motivation and contribution

WSNs are designed to sense, gather and transmit useful information to interested users or applications. In some applications, sensor networks are expected to interact with the physical world. They may respond to the sensed events by performing corresponding actions and assist people in their life, such as in Wireless Sensors and Actuator Networks (WSANs). For example, in a fire handling system, the actuators can turn on the water sprinklers upon receipt of a fire report [5]. Sensors can monitor and manipulate the temperature and lighting in a smart office or the speed and direction of a mobile robot [257]. To produce useful outcomes in a specific WSNs application, many sensors collaborate in monitoring and gathering information. In patient monitoring system, sensors can signal a possible health trouble, by gathering information from blood pressure, glucose and other monitoring body sensors. An accurate data analysis and processing is an important issue to increase the smartness and lifetime of nodes. However, data management (i.e collection, processing and communication) is not a trivial task, especially for energy-limited sensor networks. The energy of a node is consumed in three aspects: sensing, processing and communication. It has been mentioned in [192] that the amount of energy consumed by the transceiver varies about 15% to about 35% of the total energy. Nonetheless, the majority of energy is consumed in radio communication rather than in computation. For example, in Sensoria sensors[5] and Berkeley[6] motes, the ratio between communication and computation energy consumption ranges from 10^3 to 10^4 [271]. Moreover, sending 1 bit over 100 meters consumes the same amount of energy as the execution of 3000 instructions [230]. As for the transmission and reception costs, the transmission of one bit consumes twice as much energy as receiving a bit [116]. Therefore, we focus on reducing the communication cost rather than computation cost, as many approaches did [74, 218, 271]. More specifically, we tackle the transmission cost reduction.

As each node is a data source that can be equipped with multiples sensors, a lot of data can be collected, processed and transmitted, such as humidity, temperature, location values, etc. The transmission of data is mostly realized through multi-hop network architectures due to their energy-efficiency [175]. During this multi-hop transmission, the communication overhead can be increased. Additional relevant information from other sensors can be aggregated through in-network processing [118]. Aggregated data can be large, erroneous, or redundant which may harm the sender's source of energy. Therefore, observing the quality of data during aggregation can omit unnecessary communication. We aim at reducing the energy consumption during aggregation. We try to decrease the transmission rate and overhead between nodes. Our concern is to strike the balance between the transmission cost reduction and the quality of collected data from the network. We aim for a less but useful data transmissions. If data were erroneous or anomalous, based on an application criteria, and they were not detected, they will surely affect the behavior of sensors. They will decrease the network performance, since it impacts the quality of produced information/decisions, as well as the energy consumption.

In this work, we do not study the computational cost of nodes. Rather, we aim at reducing the transmission cost, during aggregation, in energy-limited sensor networks. In this thesis, we propose the following:

- A Prediction technique described in Chapter 3.
- A Data correlation and similarity detection technique described in Chapter 4.
- An Anomaly detection technique to increase the quality of sensed data, described in Chapter 5.

1.5 Organization of the Thesis

This thesis is organized in five chapters.

- Chapter 2, presents different approaches, proposed in the literature, to save energy for communication in Wireless Sensor Networks (WSNs). In this chapter, we observe data-driven techniques, such as in-network processing, data compression, data prediction, etc. These techniques have been considered energy-efficient in terms of communication [101]. Our algorithm relies on data aggregation and estimation to decrease the transmission rate and overhead.

[5] Sensoria Corporation: http://www.sensoria.com

[6] http://www-bsac.eecs.berkeley.edu/

- Chapter 3, presents our algorithms proposed for **E**nergy-**E**fficient **E**stimation (such as EEE, EEE^{+}, EEE^{*} algorithms). These algorithms aim at predicting local readings and reducing the communication overhead. We adopt a cluster-based data aggregation scheme and focus on reducing the transmission overhead between a source node and its sink (i.e the Cluster-Head). We use static and dynamic bound to observe the estimation quality. Simulations show that depending on data type, transmission overhead and rate can be reduced. We noticed a reduction about 70% in transmission rate [100]. A considerable accuracy prediction can also be obtained. Results show that the relative errors between predicted and sensed data were between 2% and 8%.

- Chapter 4, investigates spatial and temporal data correlation in order to detect redundancy and reduce data transmission overhead. We propose a data similarity detection technique that relies on kernel-based methods. Simulations show a data transmission overhead reduction of about 20%, in most used data types. While for the accuracy on the base station estimations, results reveal a good estimation quality since the relative error values between source and sink estimations were in $[-7 \times 10^{-7}, 10^{-6}]$.

- Chapter 5 presents a technique for anomaly detection in WSNs. Since Data measured and collected by WSNs is often unreliable and affected by different factors (noise, error, abnormal and inconsistent data), an anomaly detection technique is required to maintain the reliability of the network. Our algorithm produces high detection rate of abnormal behaviors. It can accurately detect between 78% up to 90% of slipped anomalies. It also produces low false alarm rate ($\sim$8%) for slow variation data types. The false alarm rate is the ratio between the number of normal values that are incorrectly misclassified as outliers and the total number of normal values.

- Chapter 6 summarizes the thesis and presents several perspectives of our work.

When you take stuff from one writer it's plagiarism; but when you take it from many writers, it's research.
∼∼∼∼*Wilson Mizner*

Chapter 2

State of the Art

Summary

The core idea of this chapter is to highlight the way data management can impact the energy consumed in Wireless Sensor Networks. We observe the transmission rate and overhead. It first discusses the terms: data management, data quality and quality metrics. Then it moves to present some data-driven techniques proposed in the literature to reduce the energy consumption. The chapter is closed with a summary of our thesis contribution.

Contents

The innovation of smart environment is increased with the development of intelligent systems related to industry, home, transportation, medicine, etc. [235]. In [51], smart environment is defined as a world where different kinds of devices are continuously working to acquire and apply knowledge about the environment and its inhabitants; they aim to make inhabitants' lives more comfortable and improve their experience in that environment. In this vision, the world around (homes, offices, hospitals, cars) is organized as a pervasive network of intelligent devices that will cooperatively gather, process, and transport information. The information from smart environment are mostly provided by Wireless Sensor Networks (WSNs) [88]. Sensory data are collected from one or multiple sensors. They can be processed in different ways. Data processing in WSNs is organized into two groups: centralized and distributed processing [243]. In the centralized processing, data are collected and sent to a central node for processing. In the distributed processing, intermediate nodes become more aware of the final results or decisions. They collect and pre-process data to obtain partial results sent to the sink. This type of processing is called node-level distributed processing. To obtain a global information, final decisions are made from the node-level output and from the information exchange between the sink and sensors as well as between intermediate nodes. This is called the network-level distributed processings.

As examples, Botanicalls[1] open a new channel of communication between people and plants. They use moisture sensors, process sensed data and take a decision to communicate with people. They may use twitter, make calls or send text messages, if they need to be watered. Plants in a garden may also communicate with sprinklers. An automated sprinkler system [83] is equipped with humidity sensors and sprinklers as actors. The sprinklers are activated when the humidity sensor readings go below a certain bound. Therefore, data are collected locally, and from surrounding nodes, and are processed to take decision for irrigation. It is preferred that only a minimum subset of sprinklers is activated to cover the entire region, so that the overall consumption of sprinkler resources (i.e water), and energy is minimized. On the other hand, data collected can be similar, redundant, erroneous or anomalous. The collection and accumulation of these data, results on spending large amount of energy during transmission, which decreases the network performance. Unwanted actions can be triggered due to low quality data gathering, and nodes' lifetime declines faster. Data fusion [164], is generally defined as the use of techniques that combine and gather data from multiple sources an energy-efficient manner, such as data aggregation, estimation and compression. The core idea of these techniques is to reduce the energy consumption in the network. The encoding schemes [11] "compress" data, in a way these data are represented by smaller number of bits. The more data are represented using fewer bits, the less is the energy required by every node to transmit these compressed data [224, 230]. Compression methods aim at presenting significant and accurate data representation, so that the original data structure can be recreated at the destination node from those few bits. Other techniques, were also proposed to reduce the communication traffic while keeping a desired level of data quality, such as outliers and redundancy detection [142]. Techniques, such as data prediction [124, 263], can reduce the number of communication between nodes,i.e. the transmission rate. Thus, communication between nodes will occur, if the tolerated bound for prediction error is

[1] http://www.botanicalls.com

crossed.

Given that the computation consumes less energy than communication (as presented in Chapter 1), we try, in our work, to reduce the energy consumption for transmission. This work relies on data aggregation, estimation and outliers detection techniques. The aim of this Chapter is to review some of the existing data-driven techniques (Section 2.3). We first identify the terms data management and quality in Section 2.1. Then we move to present some of the data quality metrics in Section 2.2. We end the chapter with a summary of our contribution (Section 2.4.

2.1 Data management in WSN middleware

WSN middleware is a set of services and supports, located between the sensor application and the networks stacks (transport, routing and MAC layers). It aims at improving the system performance and the network efficiency (see Figure 2.1) [243]. WSNs middleware acts as a broker between the applications and the network infrastructure. Some of its services are: location tracking services [33, 150, 160], coverage [43, 245], data management services such as data aggregation, routing service, cluster service [38, 148], etc. The main purpose of middleware for sensor network is to support the development, maintenance, deployment and execution of sensing-based applications. Good references on wireless sensor network middleware and services are presented in [113, 243].

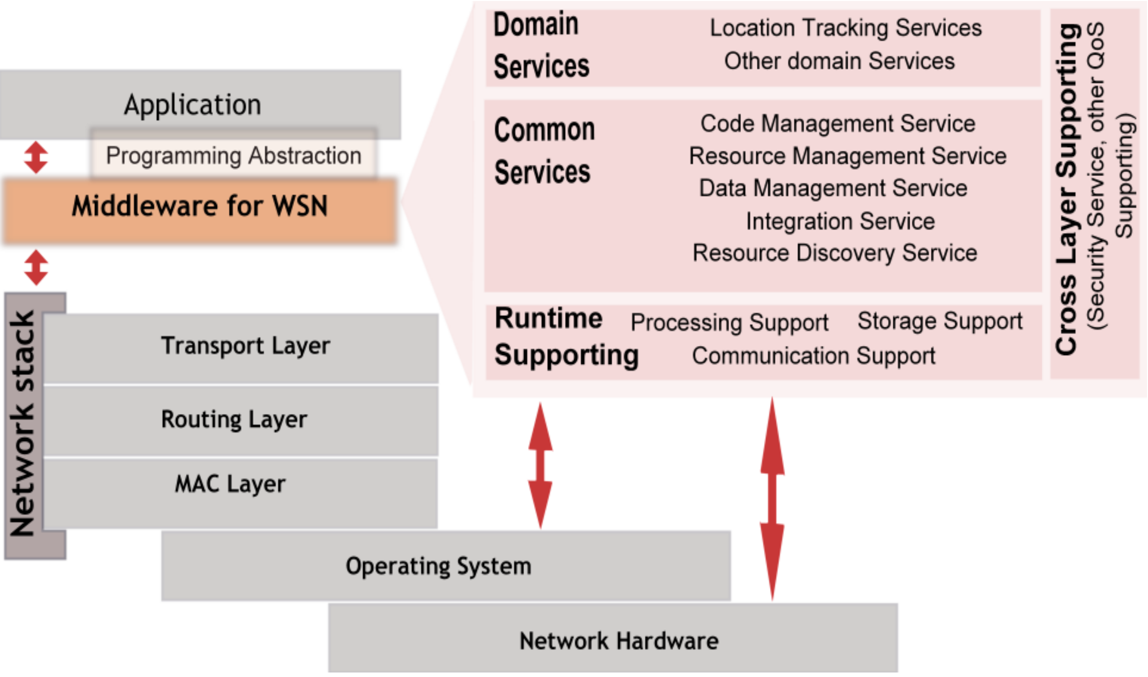


Figure 2.1: Reference model of WSN middleware.

2.1.1 Data management

The data management is one of the basic and important services in WSNs [243]. It provides services for data acquisition, processing and storage. *Data acquisition* is a service aiming at delivering the relevant and accurate data required by an application. In [212], data acquisition is stated as collecting, processing and transmitting data. It includes sampling [101], network layer routing [152, 266], data aggregation [185], etc. The way data are manipulated, is a fundamental issue [78]. It should take into consideration, the sensor node's processing capability and lifetime, as well as the application requirements (data/information quality: accuracy, timeliness, etc.) [16, 78]. Some of energy-efficient data-driven techniques are presented in Section 2.3. Data-driven, means that applications collect and analyze data from the environment, and, depending on redundancy, noise, and properties of the sensors themselves, can assign a quality level to the data [113]. The quality of data and some of its metrics as well as the metrics we have used in this thesis are discussed in Sections (2.1.2 and 2.2).

2.1.2 Quality of data

Quality is referred as the degree of satisfaction of a set of criterias. According to [177, 216], data quality is 'contextual'. The user defines what is good data quality for each proposed use of the data, within its context of use. Many applications have real-time deadline requirements for data collection, to proceed for the appropriate reactions, such as, military applications, surveillance, fire detection and intrusion detection [36, 141]. Data should be accurate and transmitted in predetermined latency boundaries. If data collected arrive beyond their validity to a central node for processing, they are not useful at all. Nonetheless, if data are erroneous or anomalous based on the application criteria, they will surely affect on the behavior and reaction of the application. For example, in a smart nursing home WSN scenario, it is necessary to guarantee life-threatening events such as heart-attacks. Information are communicated to doctors within a bounded time [94, 140, 161]. Errors in estimating a patient-state, may decrease the possibilities for rapid intervention to patient urgent-treatment.

The real concern with data quality is to ensure that data are accurate enough, timely enough, and consistent enough to make appropriate and reliable decisions [20, 177, 182, 227]. Researches in the literature [34, 180, 202, 259] have measured the quality of data in the context of their work, using different attributes: accuracy, timeliness, precision, consistency [1]. In what follows, we define each attribute then we mention some of the quality metrics in Section 2.2.

- Accuracy: It is the degree of closeness of measurements to the true value. The accuracy of a sensed value is the difference between the sensed reading itself given by the sensor and the true value from the environment. It is well known that no data is completely accurate. Since the true value is not known by a WSN applications, the true accuracy of sensor readings cannot be known. Actually, if a WSNs application knew the true value, there would be no point in taking a sample. When a quantity is measured, the outcome depends many factors, such as the measuring system, the measurement procedure, the skills of the operator, the environment, etc. [17]. On sensor level, the calibration of the device improves accuracy. Quality cannot be exactly determined, but must be assessed by means of estimation. In this work, we consider accurate, the sensed data we have used for simulations. Theses values were taken as reference to evaluate the accuracy of our proposed works.

- Consistency: It is a sub-aspect of accuracy. If a single or set of data is compliant with a user-defined model, it is consistent. A user-defined model may, for example, make an assumption between several sensor readings. It may require that for two temperature readings taken at the same location, the difference may not exceed 0.5°C. If data fulfill the requirements of the model, data is said to be consistent with the model. We rely on the model proposed in Chapter 3, to observe data consistency and detect anomalous values (see Chapter 5).

- Timeliness: It indicates whether the data reach the destination in time. It is influenced by unreliable radio-communication and network latency. Timeliness is out of the scope of this work. In the following chapters, we assume that data reach the destination in a timely manner.

- Completeness: It is a property of a set of sensor readings. If a node has taken (or received from the network) a sufficient number of samples to construct the true value, then the set of samples is complete. In Chapter 3, we observe the adequate prediction window for our estimation model. We mean by "prediction window", the number of samples used to estimate a value. Completeness is affected by errors at sensor level as well as errors during data collection such as lost or duplicated data. We assume that data reach destination, without loss, in a timely manner.

- Precision: It is the degree of reproducibility of measured values which may or may not be close to the real world value. In Chapter 3, we observe the precision by calculating the relative error for estimation, the percentage of accepted values. Values are called accepted by our model, when they agree with the user-defined criteria during the change of the window size.

While talking about the quality of data, it is important to observe the quality of information. It is affected by the quality of data used to complete information, in other words, the quality of data collection and processing. In WSNs, the Quality of Information (QoI) [21, 95, 229, 267] is significant. It indicates to what extent the network fulfills the applications needs and network concerns. QoI in regard to monitoring environment is described in [95], as "the difference between the data produced by the WSNs concerning some event being monitored, and the actual events in that environment which one wishes to observe or track".

According to [267], QoI is "the collective effort of the sensor derived information, that determines the degree of accuracy and confidence by which the aspects of the real world (that are of interest to the user), can be represented by this information".
Others, such [21], stated QoI as "a characterization of the goodness of the data captured by and following through the WSNs, and the information derived from processing those data along the way". Hence, QoI should satisfy user requirements from the time of raw data collection till the operations at the sink are conducted. To obtain information of greater quality, data need to be collected, analyzed and fused in a way to gratify applications

needs [239]. By taking advantage of the cooperative, complementary and redundant aspects of data, quality of information can be improved in WSNs [164]. Let us, briefly, describe these quality aspects of data fusion:

- Complementary: When information provided by the sources represent different portions of a broader scene, information fusion can be applied to obtain a piece of information that is more complete.
- Redundant: If two or more independent sources provide the same piece of information, these pieces can be fused to increase the associated confidence (more accurate information). Redundant fusion might be used to increase the reliability, accuracy, and confidence of the information. Whereas, redundant fusion can provide high quality information, it needs to be managed in an energy-efficient way to prevent unnecessary redundant transmission (see Section 2.3).
- Cooperative: Two independent sources are cooperative when the information provided by them is fused into new information that, from the application perspective, represents better the reality. A classical example of cooperative fusion is the computation of a target location based on angle and distance information.

A major concern for WSNs design and deployment is to balance between the required QoI and energy efficiency, to lengthen the lifetime of an application. In WSNs, different Qo* exist, starting from the data collection level to the decision making level, such as quality of event detection (QoD) [111], quality of routing (QoR) [32, 128], quality of decision-making (QoDm) [182], etc. These Qo* are mainly based on QoI and are related to Quality of Services (QoS) [16] in WSNs.

In our work, we take advantage of the collaborative aspect of sensors, to achieve efficient data collection in terms of energy consumption. We aim at reducing the transmission cost to save energy. As for the quality of data/information, we observe the estimation accuracy on both the node-level and the sink level. Nonetheless, we track the data distribution for any abnormal behaviors. Since the data collected should be useful and contain less redundancy, we tackle data redundancy in Chapter 4. We propose a similarity detection technique to exploit the correlation among the sensed data. This technique fuses the similarity information, in a way to decrease the transmission overhead, while allowing an accurate data reconstruction at the sink. Simulation results show a data transmission overhead reduction of about 20%, in most used data types (see Chapter 4). As for the data accuracy on the base station, results show a good estimation quality, since the relative error values, between source and sink estimations, were in $[-7 \times 10^{-7}, 10^{-6}]$. Concerning the detection of abnormal behaviors, experiments show the ability to detect between 78% and 90% of slipped anomalous values (see Chapter 5).
In what follows, we present some of the factors that may affect the quality of data in WSNs.

2.1.3 Errors Affecting Quality of Data

There are different types of errors that affect the data quality. There are errors related to data collection and processing and others related to the device itself [159, 228].

- Errors such as random error and systematic error, are errors at the node level [8, 222]. The main cause of random error in sensor readings is an unpredictable fluctuations in the sensor device itself. Cheap sensor devices used in WSNs usually have lower precision than more expensive devices. This leads to relatively high random error in measurements. Common causes of a systematic error are incorrect calibration or incorrect usage, such as a wrong supply voltage for a sensor device. The random error of a sensor reading *sv* is the difference between the mean over all sensor readings and the value of *sv*. The systematic error of all shown sensor readings is the difference between the mean of the readings and the true value.

- Errors related to data collection and processing can occur due to incomplete or duplicate data, or due to processing techniques, such as aggregation, compression, etc. Incomplete data can occur due to two reasons: The sampling rate and unreliable radio communication. If the sampling rate is too low, relevant changes in the measured values may not be observed, leading to an incorrect perception of the sensor reading. Nonetheless, if radio communication is unreliable, data may be lost (due to radio interference, a node cannot be reached by the data source). Another situation resembles data loss, that is outdated data. Data may be outdated if the network delay is too high. Thus, data become irrelevant once they reach the destination.
  The use of multi-path routing (Broadcast) in WSNs [165, 172], causes severe message collision and channel contention. Data may be duplicated intentionally.
  Techniques such as in-network data fusion, compression, data aggregation, are used to reduce redundancy and communication traffic. They may loose much of the original structure. This produces errors affecting the quality of data. For example, readings from a number of sensors can be aggregated into one value by taking the average over all. This value is used to represent each individual sensor reading. However, it deviates from the individual sensor readings and thus introduces an error. Also, loss of data can occur during compression techniques. Therefore, data quality is an important factor for compression algorithms [224]. It is quantified by percentage of distortion which is measured as the absolute difference between the original and the reconstructed data at the base station [2]. It is calculated as

  $$Distortion = \left| \frac{original - reconstructed}{original} \right| * 100\% \qquad (2.1)$$

  Estimation techniques also produce distortion from the true value. They aim at obtaining a close estimate of a value based on a few previous samples [124, 171, 232]. We will use this quality metric during the evaluation of our work in the following chapters.

- There are other types of errors that may occur in WSNs. They are the result of abnormal or inconsistent values [37] due to external events (fire, fraud, etc.). Such errors can change the data distribution and the behavior of an application.

In what follows, we assume that there is no data loss. In Chapters 3 and 4, we ignore errors related to the device itself. We study the aspects of redundancy and estimation in WSNs and we assume having a suitable compression and routing techniques. In Chapter 5, we try to detect abnormal behaviors in the sensed data. The main challenges in our work are: Decreasing the transmission cost, i.e, the communication overhead and rate, and maintaining adequate data quality to save energy.

2.2 Data quality Metrics

Techniques in the literature have used different metrics to evaluate the quality of data. In what follows, we present some of the statistical metrics used in anomaly detection, data correlation, compression and estimation techniques. These metrics are the Mean Squared Error [136], Root Mean Square Error [106], relative error, detection rate [226], etc.

In the following, let X_1,.., X_n be a series of n values, produced by the application that is meant to represent n corresponding true values N_1,...,N_n. The term $e_i = X_i - N_i$ is referred to as the error in the value N_i.

- ***Mean Squared error*** (MSE): is defined as the average of the squared errors.

$$e_{MSE} = \frac{1}{n}\sum_{i}^{n} e_i^2 \tag{2.2}$$

Due to the quadrature, MSE heavily weights outliers. If an application produces highly accurate values, thus for most results the error is small. However, if for a few values, errors were too large, then MSE will generally also be large. Normally, outliers have higher errors, which increase the MSE [173, 241]. Thus, the MSE can give an idea of the variance in error, but is not a good indicator of the average accuracy, if it has a large value. Another alternative to the MSE is the Root mean square error (RMSE). RMSE is defined as:

$$e_{RMSE} = \sqrt{e_{MSE}} = \sqrt{\frac{1}{n}\sum_{i}^{n} e_i^2} \tag{2.3}$$

It is a good measure of precision. The Root Mean Squared Error is commonly used in statistics for evaluating the overall quality of a regression model. It gives the standard deviation of errors. Thus, it estimates the concentration of the data around the fitted equation. In [124], the performance of the Adpative prediction model (A-ARMA), was evaluated by the root mean square error of the estimated values, with respect to, the observed ones (i.e the sensor readings).

In [259], authors also refer to the RMSE as a quality metric. They propose a data collection strategy to reduce the communication traffic. In this strategy, each sample (or value) at the sensor node is checked if it is important to be sent to the sink. On the sink side, the sink approximates the non received values by the previous samples it receives. The quality of data is measured by the root-mean-square error of the approximated data (at the sink) with respect to the sample value (i.e. sensor reading). Despite its common use, some opinions point that its use is related to its popularity with statisticians and not its efficiency in choosing accurate forecasting methods[12, 39]. They pointed that it has poor reliability in models comparison since it is not unit-free. RMSE has the same unit as the measurand (i.e. the value to be measured). Whereas, it is widely accepted that unit-free measures are necessary for comparisons among forecast methods. One of the disadvantages of the RMSE is its sensitivity to outliers as MSE does. To avoid such sensitivity, another measure has been used, that is, the Mean Absolute Error (MAE) which is less sensitive than RMSE for large errors.

- ***Mean absolute error***: is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is a common measure of forecast error in time series analysis. It has less outliers weights than RMSE, thus, it is less sensitive to large errors:

  $$e_{MAE} = \frac{1}{n}\sum_{i}^{n} \mid X_i - N_i \mid \tag{2.4}$$

  In our work, we rely on the mean absolute error of a history of predicted/true values in order to observe our model's behavior and to set the model criterias (see Chapter 3 and Chapter 5). As for the accuracy and efficiency of our work among other models, we rely on the relative error, as a metric for evaluation.

- ***Relative error***: The definition of relative error has computational advantages over the other forms and it is unit-free. It is a proportional measure of the error and provides information about the predictive capability of a model [70]. A low relative error implies good model performance and vice versa.

  $$\text{Relative Error} = (\frac{X_i - N_i}{N_i}) \tag{2.5}$$

  The relative error has been used in many approaches. In localization problems of WSNs, relative error is always used to describe the localization accuracy [246]. In [9], the estimation of the power consumption is also evaluated by this metric. In [180], it was also proposed as a query processing systems benchmark in WSNs for data quality. Due to its simplicity and popularity in accuracy measurement, we rely on this metric as a relative quantity.

- ***Other metrics:*** Techniques based on errors and accuracy are often not the best measure to detect outliers or correlation among data [56, 80, 198]. In [54, 105], data correlation techniques make use of kernel-based functions [90, 195, 268] while others make use of two popular metrics: The Detection Rate and False Positive Rate [226]. The *Detection rate (DR)* is defined as the ratio between the number of correctly detected attacks and the total number of attacks.

  $$\text{DR} = \frac{\text{number of correctly detected attacks}}{\text{Total number of attacks}} \tag{2.6}$$

  The *False Positive Rate* is computed as the ratio between the number of normal values that are incorrectly misclassified as attacks and the total number of normal values.

  $$\text{FPR} = \frac{\text{number of normal values misclassified as attacks}}{\text{total number of normal values}} \tag{2.7}$$

Often these metrics are displayed as a receiver operating characteristic (ROC) curve to emphasize the tradeoff between them [226]. We use these metrics in our anomaly detection techniques presented in Chapter 5 due to their popularity.

2.3 Data-driven techniques

2.3.1 Data/Information fusion

Many definitions has been given to data fusion in the literature. Data fusion is defined, generally, as the use of techniques that combine data from one or multiple sources. They gather information and achieve inferences, which will be more efficient and potentially more accurate than if they were achieved by means of a single source [130, 164]. In [66], it is defined as a "multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from multiple sources". It was also defined 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" [240].
Despite the difference between data and information, the terms Data fusion and information fusion are usually accepted as overall terms [64, 164]. The term information fusion was preferred to indicate the independence of a specific source of data [64]. It was stated in [164], that "information fusion is a set of algorithms, tools and methods used to process several sources of input data. They generate an output that is, in some sense, "better" then the input data individually. Saying "better" data has at least two meanings: cheaper and more accurate. In WSNs, the term "cheaper" is related to energy consumption. It means the transmission and processing of less voluminous and filtered data. "Accurate" is related to the quality of data. It means the extraction of significant outcomes, that fulfill the application requirements. We obsreve both terms in our work.

In WSNs, Sensor fusion is known also as multi-sensor data fusion. It is a subset of information fusion. It means combining sensory data/information from multiple sensors, to provide 'better' analysis and decision making, if they were done using any single sensor [75, 107, 178]. In what follows, we use the terms: data fusion and information fusion, synonymously.

Data/Information fusion provides many solutions to the energy constraints in WSNs, it uses data modeling, aggregation, time series analysis, on individual node, as well as on clusters (or groups) of nodes [104, 258]. We are interested in data fusion techniques, for battery-operated wireless sensor networks. We aim at improving energy saving during transmission, as well as, the quality of data sensed or reconstructed at the sink. Centralized or decentralized fusion refers to where the fusion of data occurs. In centralized fusion, the source nodes simply forward all of the data to a central location, where they are fused or checked for correlation. In decentralized fusion, the sources take full responsibility for fusing the data. In this case, every sensor or platform can be viewed as an intelligent asset having some degree of autonomy in decision-making [258].

In this work, fusion is centralized in the sense that it occurs on intermediate nodes, called clustrHeads. However, sensor nodes do not send all the data they receive to the central location. Knowing their weakness (limited-power resource), these nodes can make predictions and detect abnormal behaviors, to prevent periodic and low quality data communication. Based on our proposed algorithms, the decision for transmission is restricted to a tolerated prediction error bound (Chapter 3) or abnormal behavior (Chapter 5)). We try to reduce the transmission cost on one-hop, between a node and its sink. Intermediate nodes (i.e. the fusers) use our data similarity method to fuse data in a way the original data structure can be accuratly reconstructed at the destination node (Chapter 4). In what follows, we present data reduction techniques related to information fusion such as data aggregation, estimation, data redundancy, etc.

2.3.2 Data aggregation

Communication between nodes is the main source of energy consumption [147, 214]. Data aggregation has emerged as an efficient approach to reduce the number of transmissions, and minimize the overall power consumption in the network [135, 168, 185, 256]. Authors in [86], define data aggregation as "the global process of gathering and routing information through a multi-hop network, and processing data at intermediate nodes with the objective of reducing resource consumption (in particular energy), thereby increasing network lifetime".

An important aspect of data aggregation is the placement of aggregation points and how to route data to these gathering points [258]. Cooperation is particularly required between sensors to perform surveillance tasks across a region. Each sensor is expected to make its own decision about what to sense and where to transmit. But these decisions must be coordinated with other sensors to produce a global perception of the environment, and to send data to destination in an energy-efficient manner. Thus establishing a proper routing schemes makes data aggregation more efficient [242, 266].

On the other side, discovering the synergy among data can also bring significant advantages to develop efficient strategies for reducing energy consumption. In fact, the data generated by sensor nodes in the network can be *type-related.* For example, the humidity depends on the temperature. It is referred as dependencies among the attributes of the sensor nodes. They can also be *time-related.* For example, the temperature may change over time. Nonetheless, observations from the sensor nodes, which are in close proximity, may be highly correlated and the degree of correlation depends upon the distance between nodes [63, 238]. In this case, collected data are called *spatio-related.*

We suppose that sensors are sensing the same attributes. The idea is to combine the spatially correlated data (i.e. coming from different sources), and/or temporally correlated (i.e. coming from different periods of time) on the route towards the sink. Our goal is to reduce the transmission rate and overhead, thereby the energy consumption. We suppose having a proper routing protocols suitable to our techniques. In the following, we present a short overview on the data aggregation schemes in WSNs.

The routing schemes organize the sensor nodes into chain, tree or clusters. As a brief description: *Cluster-based data aggregation algorithms* organize sensors into clusters. Each cluster has a designated sensor node as the cluster head which aggregates data from all the sensors in the cluster and directly transmits the result to the sink. By grouping sensor nodes into different clusters, clustering allows hierarchical structures to be built on the nodes which can improve the scalability of multi-hop wireless sensor networks [270]. However, by selecting the cluster heads efficiently, hierarchical routing protocols can be developed to reduce the power consumption and to maximize the network lifetime. As examples, we mention the Low-Energy Adaptive Clustering Hierarchy (LEACH) approach, and the Hybrid Energy-Efficient Distributed clustering approach (HEED) [114, 264]. LEACH protocol has a randomized cluster head rotation. It assumes that all sensors have enough power to reach the sink, if needed. In other words, each sensor has the capability to act as a cluster head and to perform data fusion. This assumption might not be valid with energy-constrained sensors. While HEED, assumes multiple power levels in sensors. The cluster-head selection is based on a combination of the residual energy of each node and its degree. The degree of a node is a secondary parameter that depend on the node proximity to its neighbors. However, in both cases (LEACH or HEED), if the cluster head is far away from the sensors, nodes might expend excessive energy in communication. Further improvements in energy efficiency can be obtained if sensors transmit only to close neighbors.

Chain-based data aggregation algorithms organize sensor nodes as a shortest chain along which data is transmitted to the sink. In [138], a Power-Efficient Data-Gathering Protocol for Sensor Information Systems (PEGASIS) has been proposed. The nodes can form a chain by employing a greedy algorithm. In other words, nodes may have a global knowledge of the network or the sink can determine the chain in a centralized manner. The chain formation is initiated by the farthest node from the sink. At each step, the closest neighbor of a node is selected as its successor in the chain. In each data-gathering round, a node receives data from one of its neighbors, fuses the data with its own, and transmits the fusion result to its other neighbor along the chain. The leader node, that is similar to cluster head, transmits the aggregated data to the sink. PEGASIS assumes that all sensors are equipped with identical battery power. In addition, It results in excessive delay for nodes at the end of the chain which are farther away from the leader node.

Tree-based data aggregation algorithms organize sensor nodes into a tree. Data aggregation is performed at intermediate nodes along the tree and a concise representation of the data is transmitted to the root which is usually the sink [118, 146]. To construct and maintain a data-aggregation tree in sensor networks, an energy-aware distributed heuristic (EADAT) [73] has been proposed as well as another protocol called (PEDAP). It requires global knowledge of the location of all nodes at the sink and operates in a centralized manner where the sink computes the routing information.

Each of these data aggregation schemes has its advantage and disadvantages [85, 135, 185]. Since the energy cost of transmissions depend on the spatial distribution of nodes, this precludes the optimality of a single scheme for all network sizes. In this work, we adopt cluster-based data aggregation scheme. We observe the temporal and spatial correlation among the sources nodes. The adoption of other schemes is among our future works. In what follows, we present some data reduction techniques, which take advantage of the spatio-temporal correlations to enhance the quality of data communication. They can also reduce the communication traffic and rate. These techniques are: data estimation, data redundancy detection and outliers detection.

2.3.3 Estimation

As periodic sensor measurements rarely change over time, a receiver can often predict them. Therefore, energy-expensive periodic radio transmission of measurements can often be omitted. Data prediction has been proposed, in several works, to reduce the data traffic in WSNs. It has been used in sensor networks monitoring (temperature, air pressure, humidity or birds monitoring), location discovery, mobility tracking applications, etc.

Data collected by sensors can be considered discreet temporal series. The time corresponds to the instant when a given value is collected. The sensed data represent the values, and the time range is determined by the duration time of the query. To predict future values of sensed data, it is necessary to find a trend in the temporal series and to observe the data distribution.
Most of data prediction works rely on times series forecasting with a linear regression model [67, 69, 124, 233]. Different estimation methods use the laws of probability to compute a set of measurements based on a sequence of measurement vectors [26]. Some of these are: Maximum Likelihood, Maximum A Posteriori, Least Squares, Moving Average filter and Kalman filter.

- *Maximum Likelihood (ML)*: The likelihood function λ(x) is defined as the probability density function (pdf) of an observation sequence z, given the true value of the state x to estimate:
$$\lambda(x) = p(z \mid x) \tag{2.8}$$
The Maximum Likelihood estimator (MLE) searches for the value of x that maximizes the likelihood function λ(x). The MLE is commonly used to solve location discovery problems, such as Birds monitoring and localization [44]. In this context, the method is often used to obtain accurate distance estimations, such as direction or angle, that are used to compute the location of nodes or targets [44, 84, 171].

- *Maximum A Posteriori (MAP)*: This method is based on Bayesian theory. It is closely related to maximum likelihood (ML). However, (ML) method assumes that x is a fixed, though unknown, point of the parameter space. MAP takes x as the outcome of a random variable with prior pdf known. The MAP estimator was used to find the joint positions of mobile robots in a known environment. It is used to track the positions of autonomously moving objects [194], manage the collision and traffic between the source nodes (detectors) and a fusers (cluster-heads). MAP estimator computes the number of nodes that wish to transmit, so they can properly update their retransmission probability [265].

- *Least Squares*: It is a mathematical optimization technique. It searches for a function that best fits a set of input measurements. This is achieved by minimizing the sum of the square error, between points generated by the function, and the input measurements. As a square-error metrics, we mention the root mean squared error [106] and the ordinary squared error [30], which is suitable when the parameter to be estimated is considered fixed. The Least Squares method searches for the value of x that minimizes the sum of the squared errors between actual and predicted observations. To reduce communication, some methods [106], avoid the transmission of the complete data stream from source to sink. They choose to share the parameters of a linear regression that describes the sensor data. The values of these parameters are estimated by applying the Least Squares method with a root mean squared error as the optimization metric.

Other methods [193] use a dual prediction scheme, in both the source and in the sink, based on Least Squares filters. Only when a predicted value differs from the actual one by more than a given error bound, it is transmitted to the sink. Least square methods have been also used for node localization such in [139].

- *AutoRegressive (AR)/MovingAverage (MA)*: The moving average filter is widely adopted in digital signal processing (DSP) solutions [210]. It was used to estimate the data traffic for routing-failure detection [163], and for target locations to reduce errors of tracking applications [260].
Some approaches use a tool called AutoRegressive-Moving Average (ARMA) for forecasting [40]. This tool contains a moving average (MA) model, as well as, an AutoRegressive (AR) model. It is typically applied to autocorrelated time series data. This model predicts future values in the series. It is referred usually to as the ARMA(p,q) model as follows.

$$AR(p) : X_t = c + \epsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} \tag{2.9}$$

$$MA(q) : X_t = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{2.10}$$

$$ARMA(p,q) : X_t = c + \epsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{2.11}$$

where c is constant (often omitted), p is the order of the autoregressive part and q is the order of the moving average part. The terms ϵ_t,ϵ_{t-1}.. are referred as white noise (i.e. a random error), φ_i and θ_i are the parameters of the AR and MA models respectively. Approaches like [124, 233] use dual prediction AR/ARMA models contained in both the source and the sink models. In [233], the Probabilistic Adaptable Query system (PAQ) relies on local probabilistic models computed and maintained at each sensor. It uses an AR model as the probabilistic model. The work in [233] is designed to be lightweight. It shows that AR models, while simple (simpler then ARMA), still offer excellent accuracy in sensor networks for monitoring applications, making such models practical on many current-generation sensor networks. The sink node exploits AR model to predict local readings instead of direct communicating with sensors. The prediction model is contained in both the sink and each sensor. PAQ is highly adaptable to changes in the network, since all data are local and a priming stage is not required. This also leads to fewer transmissions between nodes, which increases its efficiency in sensor optimization. Valois in his work [124], gives nodes additional task over environmental monitoring. Every node calculates an Adaptive-ARMA (A-ARMA) for estimation. This model is re-learned when a value is mispredicted. To prevent large data communication, A-ARMA transmits the model parameters, and a sample of previous measurements instead of raw sensor readings.

The decision making for transmission, in these approaches [124, 233], is related to a prediction error bound. However, in the absence of prior information about the physical environment, setting this bound is a nontrivial task. The threshold can be used as a tuning parameter to characterize the tradeoff between accuracy and energy consumption. It can be user-defined [250] or dynamic threshold using slide window method [179, 233].

- *Kalman Filter*: It is a stochastic, recursive data filtering algorithm. It has been well studied and widely applied to many data filtering and smoothing problems. It has found applications in different fields of process control [181], multi-sensor data fusion [253], motion-tracking [249], network-time keeping [24], and neural information processing [254], etc. Kalman filter comprises a set of mathematic equations that provide a recursive solution to the least-squares method.
  Still in the context of data communication, approaches such as [122] use a dual Kalman Filter where both source and sink nodes predict the sensed value. The source node sends data only when the sink prediction is incorrect. Kalman filter was also useful in routing algorithm such as the SCAR routing algorithm [152]. A sensor node uses Kalman filter to predict context information about its neighbors (mobility and resources). It chooses the best neighbor for routing its data relying on such predictions.

  In our work, the prediction model is an AutoRegressive-based model for data prediction. It aims at predicting local readings and reducing the communication cost. The model parameters are updated during the prediction process to keep accuracy in predicted values. They are not transmitted to the sink as in [124]. Data are only communicated, when the tolerated prediction error bound is crossed. The performance of our estimation model was evaluated using static and dynamic threshold. The experimentation included comparison with A-ARMA proposed in [124]. Results show that depending on data type, transmission overhead and rate can be reduced (about 70% reduction in transmission rate [100]). A considerable accuracy prediction can also be obtained. The relative errors, between predicted and real sensed data, were between 2% and 8% (see Chapter 3).

2.3.4 Data redundancy

Redundancy in Wireless Sensor Networks can be classified into *spatial* redundancy, *temporal* redundancy and *information* redundancy [59].
Spatial redundancy means the possibility to obtain a specific information from different sources. Because nodes in WSNs are typically deployed densely [207], this provides a large amount of redundant data in network coverage. Nonetheless, the use of multi-path routing in WSNs [165, 172] may produce duplicated data.
Temporal redundancy may occur in sensing and communication. Temporal sensing redundancy is defined as obtaining multiple measurements from the same sensor, skewed in time (such as in video surveillance). Temporal communication redundancy is defined as sending the same package of data more then once, skewed in time. Such as the Time-out and retry mechanisms used by several communication protocols [3, 61].

Information redundancy occurs using of redundant data. It is linked to spatial and temporal redundancy.

In sensor networks, redundancy is both ally and enemy [59]. It is used sometimes to improve the reliability of the entire system [221]. Kalman filters are very popular fusion methods that benefits from redundant information to improve their data estimation task [142]. They have been used for a long time in different application areas, such as in algorithms for source localization and tracking, especially in robotics [142]. In biomedical data fusion, redundant and complementary information can be useful. For example, the fusion process can improve the detection of cardiac events including the ventricular activity and the atrial activity [115, 161].

However, the collection and accumulation of redundant data (similar or duplicated data), results on spending large amount of energy during transmission. This may drain the source of energy. Along these concerns, data aggregation techniques [73, 166, 172, 185], were used to minimize redundant data, thereby the transmission overhead, which prolongs the network lifetime. Although redundancy might have its negative impact on the network lifetime, however, it might be useful to increase the reliability, accuracy, and confidence of the information. Decreasing the transmission rate and traffic may produce a loss of accuracy during aggregation. Therefore, the compromise, between the desired level of redundancy in the network, the quality of aggregated data and the network lifetime, constitutes a performance measure of an efficient data aggregation algorithm [135, 185].

We tackle the issue of redundancy in Chapter 4. We propose a similarity detection technique to observe the correlation among sensed data on the aggregator node. It fuses the similarity degree, in order to decrease the transmission overhead and to allow the reconstruction of fused data at the sink. Results of simulations show a data transmission overhead reduction of about 20%, in most used data types. They also a good estimation quality, on the base station. The relative error of prediction were in $[-7 \times 10^{-7}, 10^{-6}]$.

2.3.5 Outliers detection

The quality of data (collected or communicated) in WSNs affects the performance of nodes and the whole network. If data are not accurate or reliable, actions and decisions, that will be taken based on such data, can not be reliable nor fulfilling the applications needs. Such data produce fake information and may lead nodes to unpredicted behaviors or unwanted motions which may affect the source of energy and degrade the performance of the network. An unreliable home sensor, may signal a fake fire event, which activate home sprinklers. An unreliable mobile node may take unwanted directions, it may increase the data traffic, generates fraud information to other nodes and leads the base station to make mis-informed decisions.

Raw data collected in Wireless Sensor Networks, are often unreliable and inaccurate due to noise, faulty sensors and harsh environmental effects. Such data are generally called “outliers”. Outlier detection techniques are proposed to reveal such cases and improve the performance of the network. An outlier is defined as “an observation, which deviates so much from other observations as to arouse suspicions” [110]. Barnet and al., defines outlier as “observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data” [15]. Outlier detection can be extensively used in a wide variety of applications such as fraud detection for credit card, mobile phone, insurance or health care, image processing (satellite imagery, mammographic image analysis), military surveillance for enemy activities, speech recognition, novelty detection in robot behavior, etc.

Since outliers are patterns in data that do not conform to a well defined notion of normal behavior, the key challenge is how to define normal behavior. In several cases where outliers are the result of malicious actions, the malicious adversaries adapt themselves to make the outlying observations appear like normal. Thereby, they make the task of defining normal behavior more difficult. Nonetheless, the boundary between normal and outlying behavior is often not precise. Thus an outlying observation which lies close to the boundary can actually be normal and vice versa. In many domains, normal behavior, that keeps evolving with undetected malicious values, makes the current notion of normal behavior insufficiently representative in the future.

Applying a technique developed in one domain to another is not straightforward. For example, in the medical domain a small deviation from normal (*e.g.*, fluctuations in body temperature) might be an anomaly, while similar deviation in the stock market domain (*e.g.*, fluctuations in the value of a stock) might be considered as normal. In what follows, we present some of the outliers sources in WSNs and some of the techniques used in the literature.

Sources of outlier

Outliers in WSNs can be caused by different factors: errors (noise-related measurements or a faulty sensors), duplicated data, abnormal or inconsistent data [37]. Outliers caused by errors may occur frequently, while outliers caused by events (forest fire, chemical spill, air pollution, etc) tend to have extremely smaller probability of occurrence [151]. Due to the fact that such errors influence data quality, they need to be identified and corrected in a way that data after correction may still be usable for data analysis. As events and errors are hard to be identified, outlier detection techniques need to make use of the spatial correlation between neighboring nodes for decision-making [143]. Several research topics have been developing outlier detection techniques such as fault detection [45, 143], event detection [72, 132, 151] and intrusion detection [19, 62].

Outliers Detection techniques

Different outlier detection techniques have been proposed for WSNs. Their evaluation depends on whether they satisfy the accuracy requirements while maintaining nodes energy consumption to a minimum [91]. The analysis and identification of outlier in WSNs can be made in different levels in the network. Some approaches use a local outliers detection [18, 123, 255] and others use global outlier detection [167, 205, 217]. A *local* detection is when each node identifies the anomalous values by only depending on its historical values. Another alternative is that in addition to its own historical readings, each sensor node collects readings from its neighbors to collaboratively identify the anomalous values. Compared with the first approach, the second takes advantage of the spatio-temporal correlations among sensor's data and improves the accuracy and robustness of the outlier detection [18, 123, 255]. In a *global* detection, all data are transmitted to the sink node for identifying outliers (as in a centralized architecture [205]). Data can also be sent to an aggregator/clusterhead that identifies outliers (as in clustered-based architecture [217]). In the first architecture, this mechanism consumes much communication overhead and delays the response time. The second architecture optimizes response time and energy consumption. However this approach may have the same problem as centralized one, if the aggregator/clusterhead has a large number of nodes under its supervision [37].

Different approaches for outliers detection has been proposed. Figure 2.2, presents a toxonomy of outlier detection techniques for WSNs.

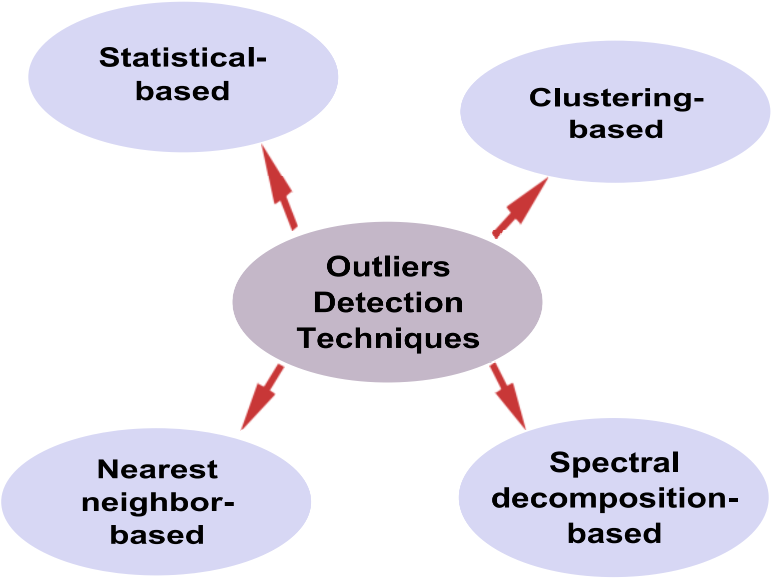


Figure 2.2: Taxonomy of outlier detection techniques in WSNs.

- Statistical-Based Approaches: They assume or estimate a statistical model. The aim is to observe the distribution of data and evaluate data instances with respect to how well they fit the model. These approaches differ based on how the statistical model is built. Among these, we mention: Parametric-based approaches (Guassian and non-Gaussian approaches) [18, 126, 255], and non-Parametric-based approaches (Kernel-based and histogram-based approaches) [167, 205]. Parametric techniques assume availability of knowledge about underlying data distribution, i.e., the data is generated from a known distribution. However, in many real-life scenarios, no a priori knowledge of the sensor stream distribution is available. Thus parametric approaches may be useless. Non-parametric techniques do not assume availability of data distribution. They typically define a distance measure between a new data instance and the statistical model. They use some kind of thresholds on this distance to determine whether the observation is an outlier.

- Nearest Neighbor-Based Approaches: They use several well-defined distance notions to compute the distance between two data instance (*e.g.* [131, 187]). A data instance is declared as outlier, if it is located far from its neighbors ([37]). Techniques, such in [273] take advantage of spatio-temporal correlations of sensor data for identifying outliers. However, a drawback of this technique, is its dependency on a suitable pre-defined threshold, that is not obvious to define.

- Clustering-Based Approaches: Cluster analysis [121] is a popular technique to group similar data instances into clusters, based on a similarity measure. Thus any data point that does not fit in any cluster is called an outlier. Since the main aim is to find clusters, such techniques are not optimized to find outliers. Other clustering based techniques have extended to time-series data [23] and other sequential data [201]. The advantage of the cluster based techniques is that they do not have to be supervised. Data points can be fed into the system and tested for outliers in an incremental way (from cluster to another). However, clustering-based outlier detection algorithms can not properly detect outliers in case of noisy data and unless the number of clusters is known in advance [261].

- Classification-Based Approaches: They learn a classification model, then classify data using two phases: The training and the testing phase. The training phase builds a classification model using the available labeled training data. The testing phase classifies a test instance, into normal or abnormal, using the learned model. Among these approaches, we mention Support Vector Machines (SVM)-based approaches [186] and Bayesian network-based approaches [123]. They can provide an exact outliers detection once the classification model is build. However, a main drawback is their computational complexity and the choice of proper kernel function.

- Spectral decomposition-Based approaches: They try to find an approximation of the data, using a combination of attributes that capture the variability in the data. Any data instance that violates the data structure is considered as an outlier. Principal component analysis (PCA) is used in several techniques such as [42] for data approximation. It is a technique used to reduce dimensionality before outlier detection. It finds a new subset of dimension which capture the behavior of data. The procedure of selecting appropriate PCA is sometimes computationally expensive [37].

Many of previous techniques [37] explore the spatio (and/or) temporal correlations among sensor data. They also differ by the outliers type they tackle (local or global) [18, 123, 167, 205, 217, 255]. Some of the oultiers detection techniques use a user-specified threshold to determine outliers [81, 188, 190, 220, 247]. Since an appropriate threshold is not easy to determine, it is usually fixed by the users. The challenge is to learn the threshold at runtime, while observing the data behavior in time. Thus, some approaches prefer to modify the threshold value when the normal data streaming is updated [269].

In our contribution for outliers detection, we use a combination of statistical and nearest neighbor technique. We define a set of ranking conditions for detection. These evaluation constraints are determined based on the previous samples of data behaviors and correlation (see Chapter 5). The experimentations we made showed that our algorithm produces high detection rate. It can accurately detect between 78% up to 90% of slipped anomalies. It also produces low false alarm rate (∼8%) for slow variation data types.

2.4 Contribution Summary

A major concern in battery-operated Wireless Sensor Networks (WSNs), is to strike the balance between the desired QoI and energy efficiency in the network. The goal is to lengthen the network lifetime and fulfill the application requirements [16, 20, 21, 32, 182]. Communication was declared as the major source of energy consumption in WSNs [6, 93]. The majority of energy is consumed in communication rather than computation[2] [116, 230]. Therefore, we are concerned, with how data sensed and aggregated by sensors can be processed to increase the performance of the networks. We observe the network performance from two perspectives: the energy efficiency and the quality of Information. We tackle the aspects of estimation, redundancy and anomaly detection in WSNs. In the following, we present the principle assumptions and headlines in our work:

- We consider that the energy consumed in sensing is negligible and we focus on the energy drained in communication. More precisely, we aim at reducing the transmission cost to save energy.

- For the sake of data consistency on node level, we consider true values, the input data we used for simulations (except, of course, the anomalous values we slipped among data for anomaly detection in Chapter 5).

- We assume that data reach the destination in a timely manner and there is no data loss.

- We suppose having a suitable compression and routing techniques.

- In this work, fusion is centralized in the sense that data fusion is done on an intermediate nodes. However, sensor nodes do not send all the data they receive to the central location. The main reason behind, is to prevent unnecessary communication and save energy. Based on our techniques, nodes have the ability to make predictions and detect abnormal behaviors. Thereby, we try to reduce the transmission cost on one-hop toward the base station, between a node and its sink.

- We adopt cluster-based data aggregation scheme and we tackle the temporal and spatial correlation among the sources nodes. The adoption of other schemes is among the future works.

- Our work is build on three related basics detailed in the following chapters: data estimation (Chapter 3), data similarity or correlation (Chapter 4) and anomaly detection (Chapter 5). Based on our proposed algorithms, nodes avoid periodic communication . They can take decision for transmission based on a tolerated prediction error bound (Chapter 3) or abnormal behavior detection (Chapter 5). The intermediate nodes (i.e. the fusers), use our data similarity method to fuse data for two objectives: Reduce the transmission cost and prevent loss of quality for the aggregated data. The goal is to enable accurate data reconstruction at the final destination (Chapter 4).

[2] Sensoria Corporation: http://www.sensoria.com

Do you realize if it weren't for Edison we'd be watching TV by candlelight?
∼∼∼∼*Al Boliska*

Sometimes a scream is better than a thesis.
∼∼∼∼*Ralph Waldo Emerson*

Chapter 3

Data prediction

Summary

Researches in Wireless Sensor Networks (WSNs) have been focusing on saving power in sensor nodes. An efficient strategy to achieve this goal is to reduce the amount of data sent through the network. The reason behind, is that communication between nodes is the main source of energy consumption [147, 214]. One of the challenges for the aforementioned goal is to exploit data correlation. Ideally, if this correlation (the true distribution of data) is known, the data transmission between nodes can be omitted. However, the true data distribution cannot be known. For this purpose, data prediction has been proposed as a way to discover the relation among data in order to reduce the data traffic. The goal is to build a model that can predict− based on known samples collected in the past− future values of some phenomenas (temperature, air pressure, humidity, etc). In this chapter, we propose a data prediction algorithm to exploit the temporal data correlation. The algorithm shows its efficiency in terms of transmission cost reduction and quality of prediction as we will present later in this chapter.

Contents

3.1 Introduction

Data collected by sensors can be considered as discreet temporal series. This collection process is called temporal aggregation. The time corresponds to the instant when a given value is collected. Periodic sensor measurements can change slowly over time, such as periodic temperature measurements, humidity, inbody glucose measurements, etc. Hence, a receiver may prefer to predict these values, so that energy-expensive periodic transmission of the measurements can be omitted. For an efficient temporal data aggregation, many techniques relied on time series forecasting with linear regression models [67, 69, 124, 233]. In this work, we propose different algorithms based on time series forecasting. The goal is to improve the performance of data aggregation in reducing the transmission cost in WSNs. We assume having a cluster-based network and we focus on the reduction of the data transmission cost on one hop communication (between a node and its cluster head).

In what follows, we provide an overview on time series forecasting, and a state of the art showing recent works applying statistical modeling techniques in WSNs (Section 3.2). Then, we describe the motivation and contribution of this work in the data prediction (Section 3.3). In Sections (3.4 and 3.5), we introduce two forecasting algorithms (EEE and EEE^+). The fundamental assumptions and setup of these prediction models are presented in Section 3.6. These algorithms were evaluated by taking a case study based on real data series and using static prediction thresholds. The efficiency regarding accuracy and energy consumption is explored in Sections (3.7 and 3.8). After comparison with other prediction models, the experimentation results show an efficiency of about 70% reduction in transmission overhead with ∼5% of relative errors. In order to enhance the prediction accuracy, we extended the work to another algorithm (EEE^*) in Section 3.9. The idea of EEE^* is to estimate a dynamic threshold for prediction errors. Since the choice of a static threshold depends on data type, a small or large static threshold can strongly provide inaccurate estimations. A tolerated prediction bound depends on the data distribution. When the distribution of data changes normally over time, the value of the bound will be affected. In the experimentations below, we consider true the data used for simulation. We assume that they do not contain abnormal behaviors so we can refer to them to evaluate our work. Abnormal behaviors will be tackled in Chapter 5.

The prediction quality of the extended algorithm was evaluated in Section 3.10 and compared to other prediction models. The experimentation was made on different real data series in several domains: The monthly measurements of carbon dioxide above Mauna Loa, the chemical process temperature readings taken every minute, daily morning temperature of an adult female, the radioactivity in the ground at 1 minute intervals over one day[1], a garden temperature data [124], and cardiac frequency measurements[2]. We will show later in this chapter, that a considerable accuracy prediction can be obtained with EEE^* algorithm. The relative errors between the predicted and real sensed data were between 2% and 8% depending on the data type. To conclude, the chapter is summarized in Section 3.11.

[1] http://www.robjhyndman.com/TSDL/

[2] http://www.inrialpes.fr/Xtremlog/

3.2 Forecasting techniques

To improve the performance of data aggregation, Time series forecasting was proposed as a means to reduce the amount of communication between the sensor and the sink. The idea behind is that the sink exploits time series model to predict local readings instead of direct communication with sensors. In this section, we provide an overview on time series modeling and some related works in the literature.

3.2.1 Modeling time series

A time series is a set of observations X_t, each of which is recorded at time t, representing a phenomena evolving over time. An important part of the time series analysis is the description of an efficient data prediction model. The prediction of future values rely on some recent history of readings. AutoRegressive (AR) and AutoRegressive Moving Average (ARMA) are principal models for time series forecasting [29, 40]. The AutoRegressive model (AR) is simpler than ARMA model due to its lower computational cost and memory requirements [29]. It becomes popular in many domains (such as in finance, communication, weather forecasting,etc.). The work in [233] shows that AR models, while simple, still offer excellent accuracy in sensor networks for monitoring applications, making such models practical on many current-generation sensor networks.
A time series model (AR) of order k is represented as follows:

$$AR(k) : X_{i+k} = a_0 + a_1 X_{i+k-1} + \cdots + a_k X_i \tag{3.1}$$

a_1,...,a_k are the model parameters. X_t is an observation of the data series at time t. a_0 a white noise (i.e. random error).
We will adopt this model due to its simplicity, which leads to lower computational cost and memory requirements unlike the general ARMA models.

3.2.2 Related works

Different techniques were introduced in the literature for data prediction in WSNs [67, 69, 122, 124, 233]. They differ by the prediction tool they use to explore the correlation among data and reduce communication cost. Some approaches rely on Kalman filters [122] or use relatively complex probabilistic models (*e.g.*, multi-variate Gaussians [68] or generalized graphical models). Other approaches [124, 233] use AR/ARMA models contained in both the sink and each sensor. By modeling time series, the sink node exploits a time series model to predict local readings instead of direct communication with sensors.

The work in [124], gives nodes, additional task over environmental monitoring. Every node, has to calculate an Adaptive-ARMA model (A-ARMA) from a history of samples to discover the time series correlation among measurements. Although it uses ARMA, the work proposed in [124], is close to ours since it uses recent readings to predict future local readings. When a reading is not properly predicted by the model, the sensor choose to re-learn that model. Then notifies the sink by sending new model parameters and a sample of recent readings. Predictions would then begin again, and continue, until an error tolerance is violated.

Other works such as [233] were more simpler. The framework relies mostly on local probabilistic models computed and maintained at each sensor. This work uses the time series forecasting model as the probabilistic model. It shows that AR models, while simpler than ARMA, provide accurate estimations in monitoring applications. Similarly to our work, each sensor continuously maintains its local model, and notifies the sink only of significant changes. However, this work differs by the cost of its learning phase. It solves a linear system to compute the model parameters. Authors refer to the standard linear algebra text [50] for this computation. However, in our model, we use an iterative method to compute the model parameters. Iterative methods produce less computational cost [108]. The method tends to adapt to errors at each iterate until a sufficient accuracy is achieved (see Section 3.4).

Moreover, in most cases, the information sent to the sink in [233], are the model parameters and optionally a list of measures. We don't send the list of parameters and/or a list of measures. We reduce the transmission overhead by only sending a list of previous errors to update the model (see Section 3.5). We will discuss the performance of our model in Sections 3.9, 3.8 and 3.7.

3.3 Motivation and Contribution

Battery operated sensor nodes are energy-constrained and their battery may be difficult to replace every time when consumed: such as in implantable body sensors, disaster or battle field monitoring sensors, etc. It has been mentioned that the majority of energy is consumed in the communication rather than computation [192, 271]. Moreover, it was indicated by empirical studies [92, 147, 214, 230] that the transmission of one bit over 100 meters would cost about the same level of energy as executing 3000 instructions in sensor node. Therefore, we aim at reducing the energy consumed in communication, more precisely, decreasing the transmission cost.

The goal of this chapter is to reduce the transmission cost between neighboring nodes on one hop communication. For this purpose, we rely on data prediction. The challenging issue in prediction is to explore the temporal correlation among data and ensure accurate predictions. In what follows, we declare the prediction scenario in our work.

Prediction scenario: Nodes monitor the same event. Each node has its local prediction model. Basically, a sensor communicates with its sink once the sensed value is mis-predicted. In other words, when a violation of a prediction error bound occurs. Regularly, when a sensor collects a new observation, it computes the error value e between this new observation and the predicted value from the model. If the prediction error becomes bigger than some pre-specified error tolerance, the prediction is not acceptable. Hence, the model risks to diverge from the data distribution. Thus, the sensor node re-computes the parameters of the model from a recent data sample. It also notifies the sink about the occurred changes, so that the latter can re-learn its prediction model. In most cases, this notification includes the model parameters and optionally a list of measures such as in [233] and [124]. In this work, we reduce the transmission overhead by sending only one previous measure (Algorithm 1 denoted by EEE) or a list of recent errors (Algorithm 2 denoted by EEE^+) [100] for more accuracy and energy-saving. We investigate the efficiency of EEE^+ algorithm in terms of communication traffic and energy consumption.

Note that, we consider true values, the data used for simulations. More precisely, we assume that data series do not contain anomalous values. Later in this chapter, we extend EEE^{+} to EEE^{*} algorithm for a dynamic prediction bound calculation. Then we discuss its efficiency.

3.4 *EEE* Algorithm

Our model is an AR-based model that uses a sequential way to estimate the model parameters, unlike other approaches that use Yule-Walker method [29] or the least squares regression [233]. Our model is based on eq.((3.1)) and is initialized as follows:

$$\begin{cases} a_0 = 0, a_i = \frac{1}{k}\ for\ i = 1, ..., k \\ X_0 = a_0 + \sum_{i=1}^{k} a_i \times N_0 = N_0 \\ X_{i+k} = a_0 + a_1 X_{i+k-1} + \cdots + a_k X_i \end{cases} \tag{3.2}$$

where N_0 represents the real value at a time instant $t = 0$. Our algorithm is based on the following idea: When a sensor detects a threshold violation, re-learning the model becomes necessary to prevent divergence. Thus, communication with the sink occurs for notification. The sensor sends the most recently obtained observation to the sink. The latter uses its previous predictions along with this observation to update its model.
This algorithm reduces the transmission overhead of the AR model part in [124]. It only sends one observation N_i at each re-parametrization step instead of $k+1$ model parameters sent in [124]. The re-parametrization step is as follow for $k = 3$:
Notations:

- N_i is the observation at time $t = i$.
- X_i is the prediction that crosses the threshold.
- e_i=N_i-X_i is the produced error at time i.
- $(a_i)_{i \leq k}$ is the sequence of parameters for the prediction model.
- $a_i^{'}$ is the adjusted value of a_i.

The new parameters of the model are computed sequentially (one after another). Every time, a parameter $a_i^{'}$ is adjusted, the value of X_i is increased by $\frac{e_i}{k+1}$. For example, after the adjustment of the j^{th} parameter ($j \in [0, .., k]$) of the sequence, X_i becomes:

$$X_i + \frac{e_i}{k+1} = (N_i - e_i) + \frac{e_i}{k+1} = N_i - \frac{je_i}{k+1}. \tag{3.3}$$

In other words, for $k = 3$, the re-parametrization of the model will start by adjusting the 3^{rd} parameter a_3 as follows:

$$a_0 + a_1 X_{i-1} + a_2 X_{i-2} + a_3^{'} X_{i-3} = X_i + \frac{e_i}{k-1} \tag{3.4}$$

We use the adjusted $a_3^{'}$ to compute $a_2^{'}$. Thus, X_i will be increased by $\frac{2e_i}{k+1}$ and so on. Generally, $a_j^{'}$ is adjusted as:

$$a_j^{'} = \frac{1}{X_{i-j}} \left[N_i - \frac{je_i}{k+1} - (a_0 + \sum_{t=1}^{j-1} a_t X_{i-t} + \sum_{p=j+1}^{k} a_p^{'} X_{i-p}) \right]. \tag{3.5}$$

It is important to notice here that this learning phase has very low computation complexity compared to the classical computation methods in AR/ARMA models (i.e. Yule-Walker or the least squares regression methods) [29, 233]. Because iterative methods produce less computational cost [108]. Note also that our model re-learning is efficient regarding communication overhead as we will show in Section 3.7.

3.5 EEE^{+} Algorithm

The main idea of this algorithm is motivated by previous energy optimization experiments showing that the energy consumed for transmission is non negligible [92, 192, 271]. Two main objectives for this algorithm are:

(1) Decreasing the transmission overhead (i.e. the number of bits transmitted to the sink).

(2) Preserving the prediction accuracy.

Therefore, we decided to send the error value e_i instead of the observation N_i sent in EEE. Note that, the sink can easily calculate the value of N_i once it receives e_i ($e_i = N_i - X_i$). By following the same adjustment procedure in EEE, the j^{th} parameter will be computed as follows:

$$a_j^{'} = \frac{1}{(X_{i-j})}[X_i + \frac{(k+1-j)e_i}{k+1} - (a_0 + \sum_{t=1}^{j-1} a_t X_{i-t} + \sum_{p=j+1}^{k} a_p^{'} X_{i-p})]. \tag{3.6}$$

Till this stage, the sink adjusts its model relying on its previous predictions. These are the accepted values, i.e. their produced prediction error do not cross the specified prediction bound. However, the sink may perform accurate model adjustment, if it receives the previous prediction errors. Therefore, to make accurate adjustment and decrease as possible the divergence of the model, a sensor node sends the recent error $e_i = N_i - X_i$ as well as the k previous errors. In other words, it sends the $(k+1)$ recent errors. Thus, a sink can accurately update its prediction model, since it can easily calculate the previous N_i observations from e_i. Thus, better quality for estimations can be generated, which will decreases the rate of re-parametrization. Thereby, energy consumption due to such transmission rate can be omitted (as we will show in the coming sections).

3.6 Preliminaries-Choice of parameters

Our concern is to reduce the transmission cost on one-hop communication. We rely on cluster-based data aggregation mechanisms in which each node can reach its corresponding sink (*e.g.* Cluster Head) directly in one-hop (such as In/On-Body sensors for personal

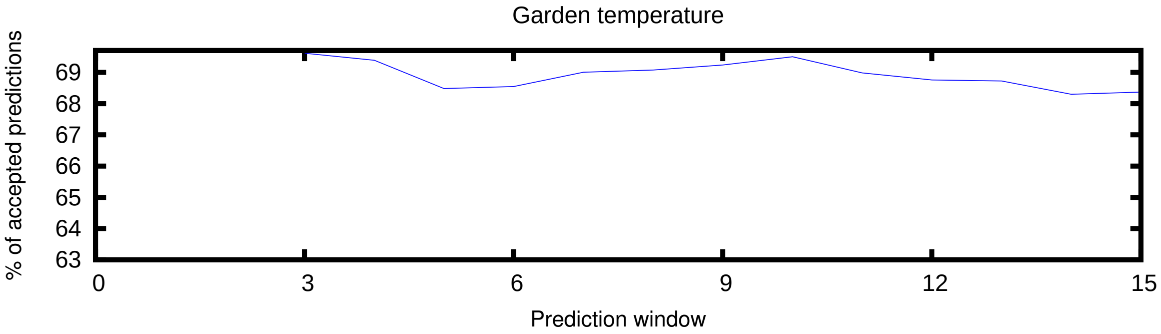


Figure 3.1: Influence of the prediction window size on the estimation precision using EEE^{+} and $th_{err} = 0.05$.

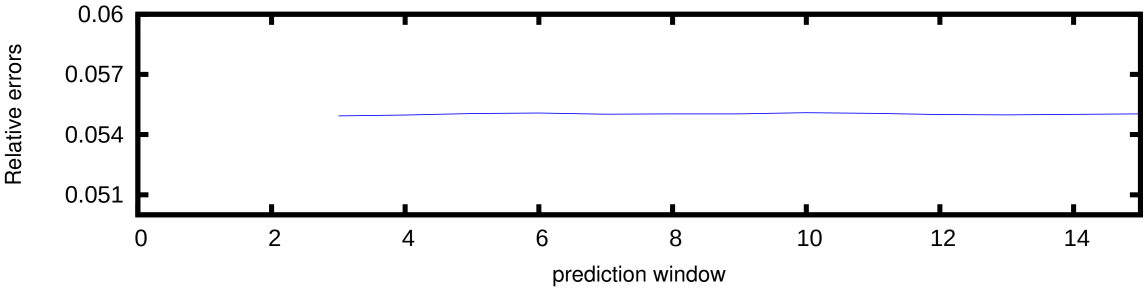


Figure 3.2: Influence of the prediction window size on the estimation error using EEE^{+} and $th_{err} = 0.05$.

health monitoring,..). We assume having a proper routing protocol. We employ an AR-based model for data prediction to prevent unnecessary periodic communication. The model is contained in both: the sink and each source sensor node. As [232], we ignore the trend and seasonal components of the time series and we set a narrow prediction window, denoted by $k = 3$ in order to decrease the complexity of learning and adapting the model. The choice of a narrow prediction window is made for two reasons: (1) to simplify our model similarly to [232, 233], (2) Our experiment results, depicted in Figure 3.1, show that a large prediction window does not really increase the percentage of accepted values. We mean by accepted values, the values that produce a prediction error respecting the bound. In fact, a large window takes into account very old information that may have changed, so the model reaction to the current change may be too slow. This could explain the deviation in the number of accepted values when the window size increases. Thereby, the rate of re-learning increases which rises the communication rate. Moreover, Figure 3.2 shows that the percentage of relative errors is limited to ∼5% while changing the prediction window size for this data type. Consequently, an increased window size will rise the communication rate, whereas similar percentage error is obtained with a small window size. Therefore, for this data type, we consider a value of 3 a reasonable compromise.

3.7 EEE^+ versus EEE and AR/ARMA

In this part, we compare the efficiency of our algorithms and different other models (such as classical AR/ARMA and A-ARMA). Table 3.1 shows the performance of the prediction models in terms of communication overhead using different threshold values. We notice that the number of mis-predictions is fewer for EEE and EEE^+ than classical AR and ARMA models. In other words, AR and ARMA models have higher transmission rate. Nonetheless, Table 3.1 shows their total communication overheads. These models generate large amount of data when the prediction bound is crossed. The low efficiency in prediction, for classical AR(3) and ARMA(3,3), is due to lack in their model adjustment. Actually, a prediction model that has been build from too old values may not remain accurate in the future. The weight given to recent values may no longer be representative for the data distribution. Thus, these algorithms lack for accuracy.

Models	Threshold	Accepted val.	Rejected val.	Overhead	Total Over. (in bits)	$r = 16$ $m = 64$ (in bits)
AR(3)	0.05	14 (1%)	1015	4 meas.	4060*m	259840
	0.1	26 (2.5%)	1003	4 meas.	4012*m	256768
ARMA(3,3)	0.05	3 (0.2%)	1023	7 meas.	7161*m	458304
	0.1	5 (0.4%)	1021	7 meas.	7147*m	457408
A-ARMA(3,3)	0.05	63 (6.1%)	963	7 param.	2247*m	143808
	0.1	135 (13.1%)	891	7 param.	2079*m	133056
EEE	0.05	668 (64%)	364	1 meas.	364*m	2912
	0.1	169 (16%)	863	1 meas.	863*m	55232
EEE^+	0.05	717 (70%)	313	$k+1$ err.	313*r*(k+1)	2504
	0.1	816 (79%)	214	$k+1$ err.	214*r*(k+1)	13696

Table 3.1: Comparison between different models using different threshold values.

The performance of the Adaptive-ARMA(3,3) proposed by [124] was also observed. A-ARMA(3,3) has the ability to adjust the prediction model from time to time. Therefore, it maintains a prediction accuracy level that is better then classical AR/ARMA. However, It has larger window size than EEE and EEE^+. Therefore, it produces less accuracy (as shown in Figure 3.3). As for the communication overhead, Adaptive-ARMA(3,3) sends 7 parameters at each re-learning phase. So according to results shown in Table 3.1, A-ARMA(3,3) has heavier communication overhead than our algorithms. We can also see from Table 3.1 that the prediction accuracy using EEE^+ is better than using EEE. This is related to the use of $(k+1)$ recent errors to re-learn the model instead of 1 measure in EEE.

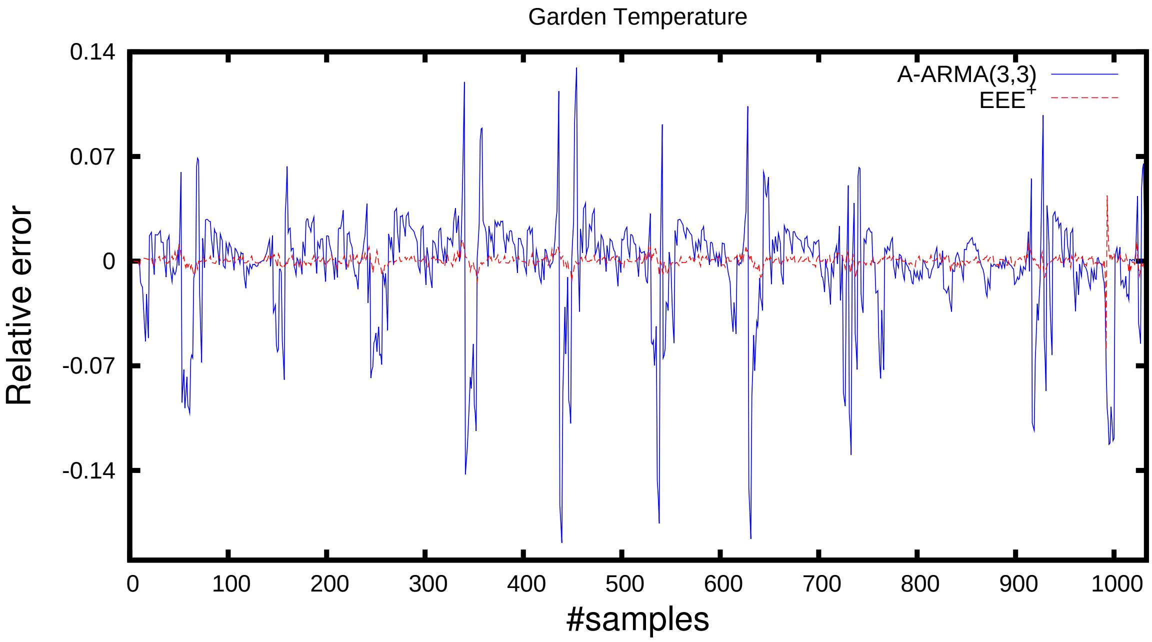


Figure 3.3: Relative error produced using EEE^+ and Adaptive-ARMA(3,3) with $th_{err} = 0.05$.

As for the prediction bound value, we notice its influence on the prediction efficiency for the different models. In fact, a large threshold value, can drive a model to divergence, since it cumulates imprecisions with the predictions. However, a small threshold value causes more model adjustments. This means more reactivity to changes and greater possibility to produce good estimations. Indeed, the selection of an appropriate prediction bound is a critical aspect, since it's influenced by the data type as well as the desired level of precision. We return to this idea in Section 3.9, where we propose a way to generate dynamic prediction bound related to data variation.

In this section, we observed the behavior of our algorithms in terms of accuracy and communication overhead. We noticed that EEE^+ produces better outcomes. Therefore, we continue to observe its energy efficiency facing other models.

3.8 EEE^+ energy efficiency

In WSNs, each node dissipates energy every time it needs to achieve a data analysis or transmission operation. For the study of the energetic transmission consumption, we consider the following simple model [244]:

$$E = E_d * b + \gamma \tag{3.7}$$

where E is the energetic consumption for the transmission of b bits, E_d the transmission energy of one bit to distance d and γ is an additional energetic consumption constant related to the transceiver and to the network packet header. If we assume that sensors use their maximum transmission power, we can set $E_d = E_+$.

Therefore for each erroneous estimation, the energy consumption (depending on the overhead type's value) is:

$$E^m = E_+ * m + \gamma \tag{3.8}$$

$$E^{kr} = E_+ * (k+1) * r + \gamma \tag{3.9}$$

$$E^{k1} = E_+ * (k+1) * m + \gamma \tag{3.10}$$

$$E^{2k1} = E_+ * (2k+1) * m + \gamma \tag{3.11}$$

Let us consider that a measure is composed by m bits and an error is composed by r bits with $r < m$. E^m is the energy consumed by sending one measure (such as in EEE). E^{kr} is the energy consumed by sending $(k+1)$ errors (such as in EEE^+). E^{k1} is the energy consumed by sending $(k+1)$ measures (such as in AR). E^{2k1} is the energy consumed by sending $(2k+1)$ measures/parameters (such as ARMA and A-ARMA).

To compute the energy consumed by each algorithm, we use the results from [112], where an energy consumption per bits is given for the CC1100[3] transceiver. The results of [112] show that we have $\gamma \sim 896\mu J$, and the transmission of one data bit is $E_+ \sim 22\mu J$. Based on Table 3.1 with $th_{err} = 0.05$, the energy consumption of EEE is $E^{EEE} = E^m \times 364$ and is $E^{AR(3)} = E^{k1} \times 1015$ for AR(3). We thus have:

$$E^{EEE} < E^{AR(3)} \tag{3.12}$$

We can notice here that $E^{AR(3)} < E^{ARMA(3,3)}$ and if we assume that $(k+1).r < m$ then $E^{EEE^+} < E^{EEE}$. One way to have $(k+1).r < m$ is to reduce the number of bits needed for the error e_i by reducing its value and choosing an appropriate coding scheme.

In Section 3.9, the estimation accuracy is enhanced using dynamic threshold which help to have $(k+1).r < m$. Results in Table 3.1, show that A-ARMA(3,3) is re-learned 321 times with a communication overhead of 7 parameters every time. This means, it is very much heavier than EEE^+ in terms of communication traffic (if we consider $(k+1).r < m$). We thus have $E^{EEE^+} < E^{A-ARMA(3,3)} < E^{AR(3)} < E^{ARMA(3,3)}$.

In the following, we consider EEE^+ an adaptive algorithm for its efficiency in terms of energy consumption and accuracy.

[3] http://focus.ti.com/docs/prod/folders/print/cc1100.html

3.9 From EEE^+ to EEE^*

3.9.1 Why EEE^*?

The prediction model requires training and re-training to maintain the prediction accuracy. Thus, the threshold value constitutes a constraint that determines when the adjustment is needed. The rate of adjustment depends on the non linearity of the data and the acceptable error tolerance. In some studies, readings are omitted, neglected or considered important when they cross certain threshold. Some events are detected using user-defined threshold [250] or dynamic threshold using slide window method [179, 233]. In fact, since data may come from different application domain, a small static threshold value can strongly increase the number of adjustment which is not energy efficient. Similarly, a large static threshold value may provide inaccurate estimation. In this section, we extend EEE^+ to EEE^* trying to estimate the threshold value at run time. Later in Section 3.10, we examine its efficiency on real data in different application domains.

3.9.2 Threshold estimation

Since the threshold is a constraint for accuracy and data calibration, we propose to calculate its value (denoted by th_{err}) as follows. We assume that a sensor sends the following values e_0, e_1, e_2, e_3 to the sink (according to EEE^+). The threshold is calculated as the mean of differences between the recent data samples N_0,N_1,N_2,N_3 (for k=3). That is $th_{err} = \sum_{i=1}^{k}(\frac{|N_i - N_{i-1}|}{k})$. Note that the choice of an error bound presents a trade-off between accuracy and error probability (ability to meet a specified confidence bound). Moreover, the estimation value is driven by a random process called white noise [60]. Some approaches like [233] consider a threshold tolerance with maximum bound equals to $\nu.b(\tau)$. $b(\tau)$ is the white noise's standard deviation and ν a real-valued constant larger than 1. In this work, we take into account the uncertainty of the estimation due to data dispersion and variability. To add some reliability to the threshold estimation, we choose to add (to th_{err}) a random value $r_{rand} \in [\frac{-c\sigma}{\sqrt{k}}, \frac{c\sigma}{\sqrt{k}}]$. σ is the standard deviation of the differences between the latest data samples N_i. c indicates the level of uncertainty (for a confidence interval of 95% we choose $c = 1.96$). To observe the behavior and deviation of the produced dynamic bound, we relied on the static bound th_s. th_s is a static threshold, computed as the standard deviation of the differences between all the consecutive real values of a data series. We generate results using different real data types: The monthly measurements of carbon dioxide above Mauna Loa, the chemical process temperature readings taken every minute, a garden temperature data [124].

	Garden Temperature	Chemical process (T°)	carbon dioxide
% of $th_{err} < th_s$	81%	73%	50%
% accept. data (th_{err})	74%	78.9%	55%

Table 3.2: The percentage of accepted estimations.

Table 3.2 shows the percentage of the dynamic th_{err} values that are less than th_s. It also indicates the percentage of accepted estimations. We notice that, depending on data type, the dynamic thresholds are, in most cases, narrower than th_s. Nonetheless, with these narrower values, there are 50% up to 80% of accepted estimations. This means that the accepted predictions may be accurate and such dynamic threshold may be useful. To observe such efficiency, we move to examine the performance of EEE^* facing other models and different data types in the following section.

3.10 EEE^* model efficiency

Performance of the models is statistically evaluated in a panel of different tables such as Data point statistics (RMSE, correlation coefficient,..), Relative error statistics (Mean value of the relative error,..), etc. To evaluate the performance in estimation of our model versus AR/ARMA models, we use: The relative error measure between the real values and the estimated ones, the correlation coefficient and the scatterplots.

3.10.1 Prediction Accuracy

Information about the absolute error is little use in the absence of knowledge about the the quantity to be measured. So, we choose to use the relative error between the estimated and the real values, as follows: $\frac{N_i - X_i}{N_i}$. We try to observe if the estimation error produced is too high or accepted according to the dynamic prediction bound. Figures 3.4 and 3.5, shows the relative error obtained using EEE^* and different other prediction models such as AR(2), AR(3), ARMA(3,3), on real data series such as: the monthly measurements of carbon dioxide above Mauna Loa, the chemical process temperature readings taken every minute, daily morning temperature of an adult female, the radioactivity in the ground at 1 minute intervals over one day[4], a garden temperature data [124], and cardiac frequency measurements[5].

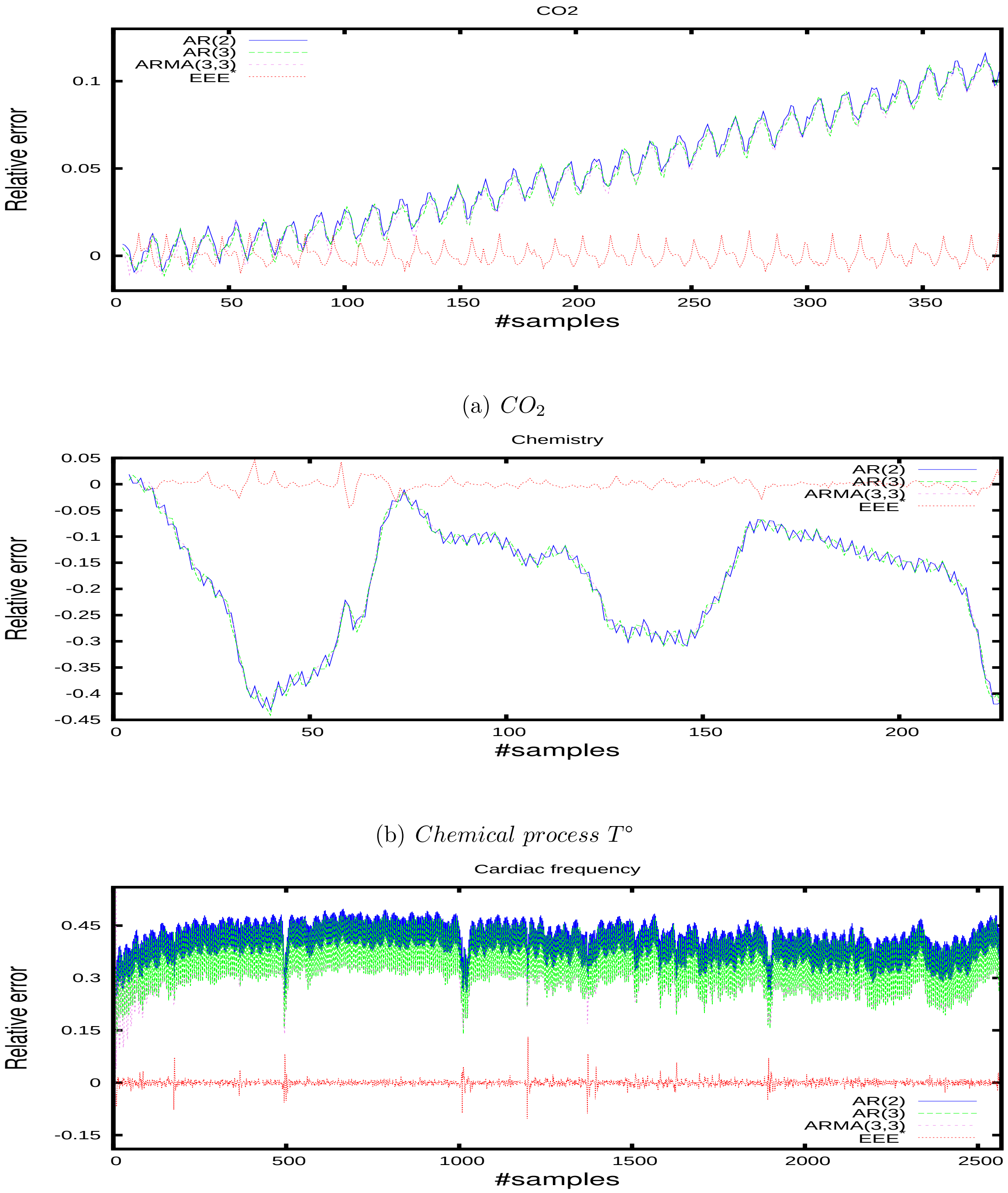
Data type	AR(2)	AR(3)	ARMA(3,3)	EEE^*
Garden (T°)	up to 43%	up to 50%	Divergence	up to 7%
Chemical process (T°)	up to 40%	up to 43%	up to 41%	up to 6%
Carbon dioxide	up to 10%	up to 11%	up to 11%	up to 2%
Cardiac frequency	up to 50%	up to 50%	up to 35%	99% of errors were $\leq$ 5%
Radiosity	up to 60%	up to 60%	up to 60%	99% of errors were $\leq$ 8%
Female (T°)	∼5%	∼5%	∼4%	∼3%

Table 3.3: The relative error produced by each model estimations: AR(2), AR(3), ARMA(3,3) and EEE^*

We generate results from AR/ARMA models using different estimation methods such as MLE, Burg, OLS and Yule-Walker. These methods produced similar results. The

[4] http://www.robjhyndman.com/TSDL/

[5] http://www.inrialpes.fr/Xtremlog/



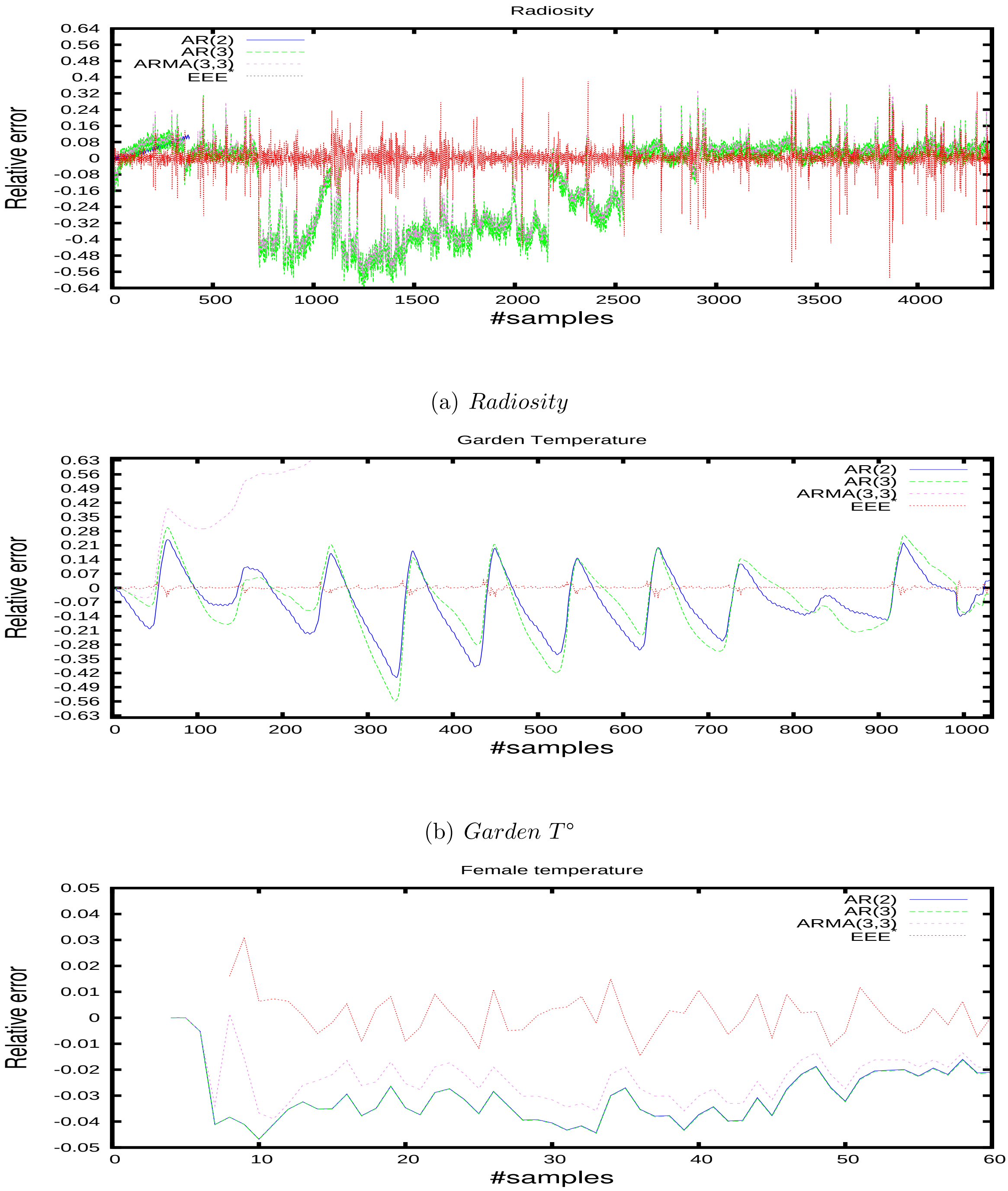
(a) CO_2

(b) *Chemical process* $T°$

(c) *Cardiac Frequency*

Figure 3.4: Relative error produced by EEE^* and different other models such as AR(2), AR(3) and ARMA(3,3) applied on some real time series data.

models did not produce better relative errors compared to our prediction model. Figures 3.4(a) and 3.5(b) show that AR/ARMA models tend to be divergent in some cases.



(a) *Radiosity*

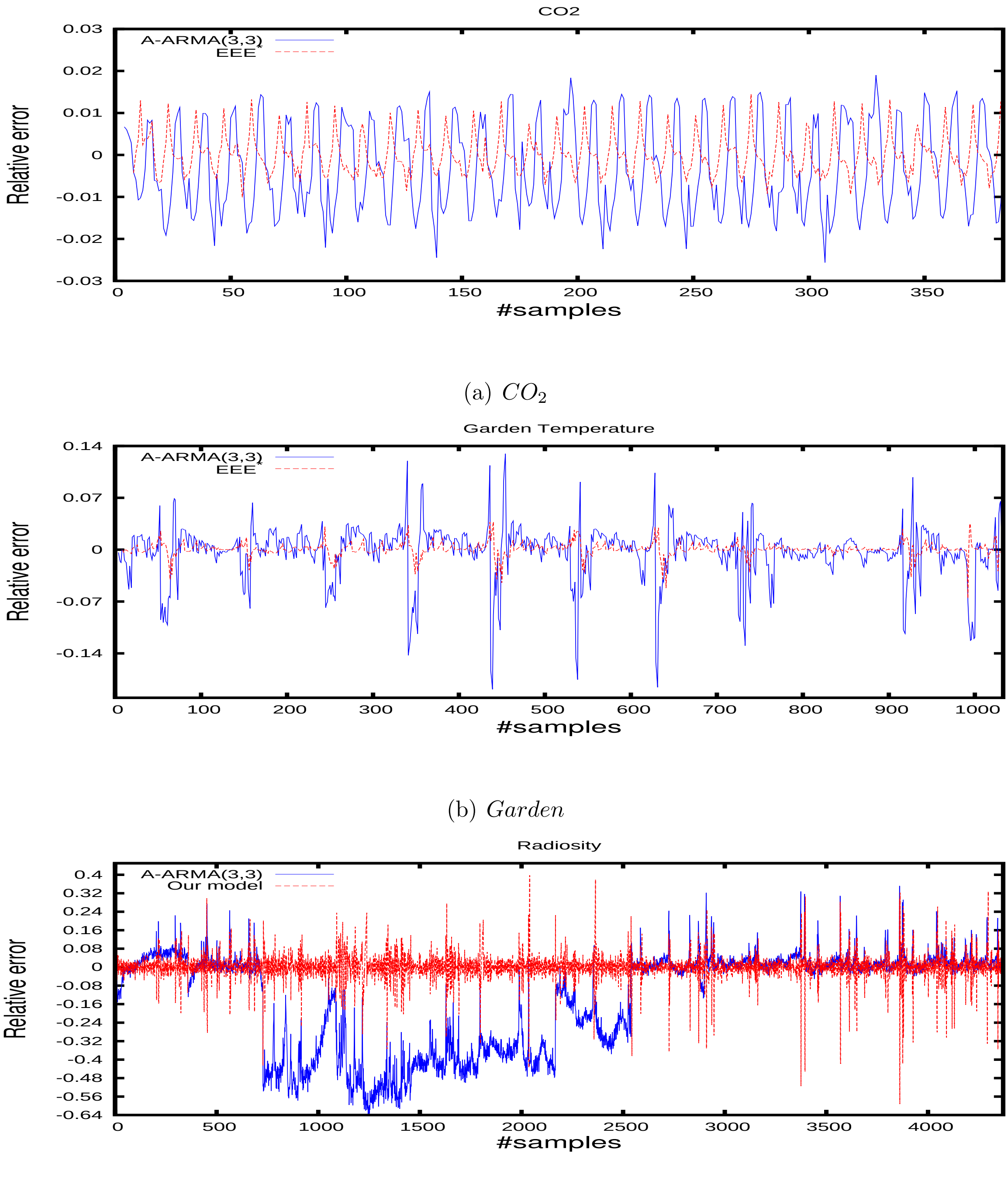
(b) *Garden T°*

(c) *Female T°*

Figure 3.5: Relative error produced by EEE^* and different other models such as AR(2), AR(3) and ARMA(3,3) on additional real data types.

This is due to the effects of looseness of accuracy along the prediction process.

As shown in Table 3.3, the relative error produced by EEE^* did not exceed 8% for



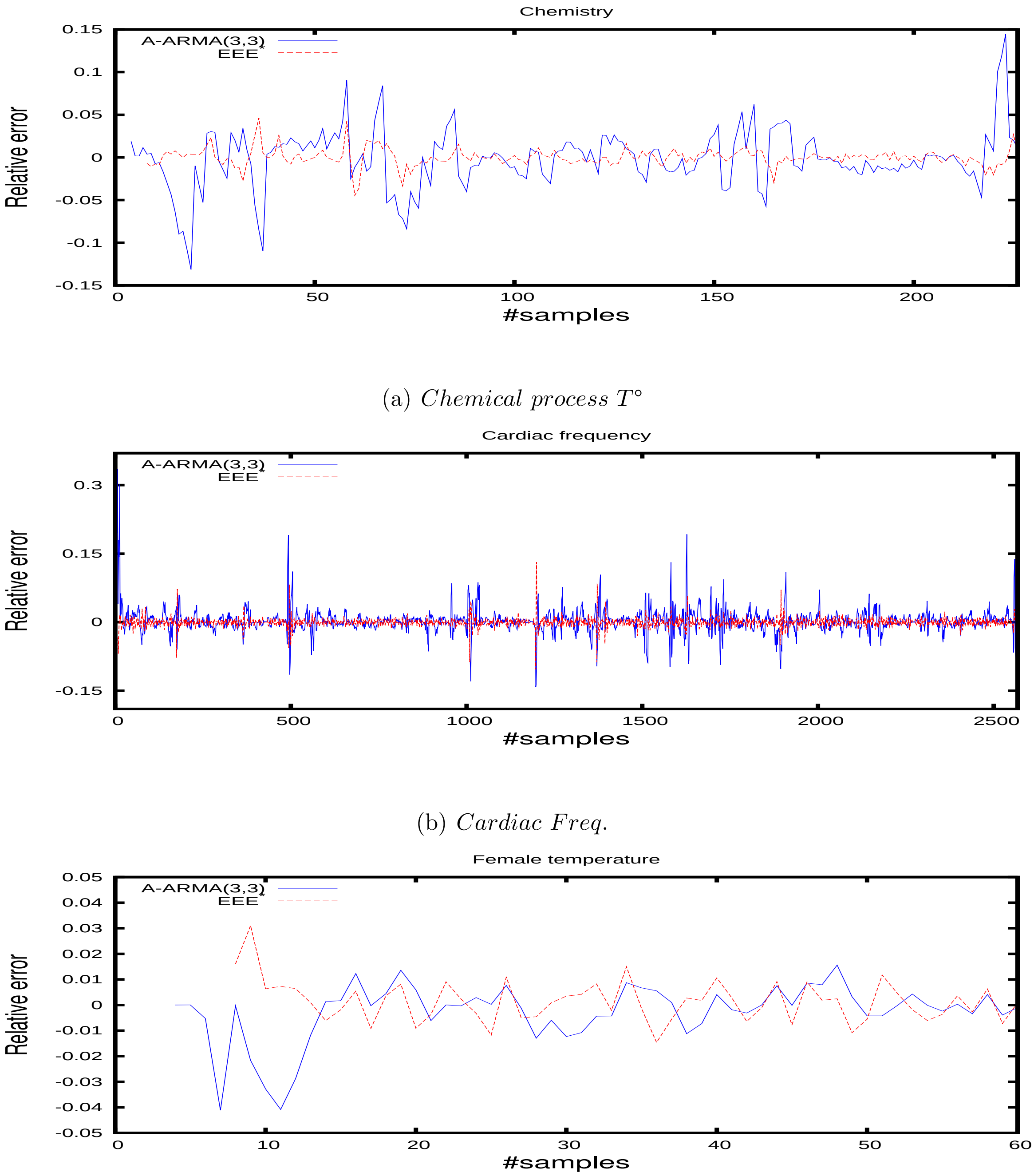
(a) CO_2

(b) *Garden*

(c) *Radiosity*

Figure 3.6: Relative error produced by Adaptive-ARMA(3,3) with a bound=0.05 and EEE^* applied on some data types.

most data types. In cases such as cardiac frequency, we notice that 99% of errors were $\leq$ 5%. Note that few errors (about 12 of 2565 predictions errors) attained 15%. We consider that this could be perturbed by different factors (sudden emotions or actions, etc..). Note that in this section, we do not detect or take into consideration such events or even data



(a) *Chemical process T°*

(b) *Cardiac Freq.*

(c) *Female T°*

Figure 3.7: Relative error produced by Adaptive-ARMA(3,3) with a bound=0.05 and EEE^* applied on additional data types.

intrusions that may occur in other WSNs applications. However some of these behaviors will be tackled in Chapter 5.

Figures 3.6 and 3.7, show the relative error produced by Adaptive-ARMA(3,3) proposed in [124] and EEE^* algorithm applied on the different data types. For the sake of fairness, we used A-ARMA(3,3) with a bound=0.05. We can notice that A-ARMA(3,3) yields the predictions better then classical AR/ARMA models, due to its model adjustment during the prediction process. However, it produces less accuracy then EEE^*. Therefore, as for the accuracy and data transmission traffic, we deduce that EEE^* is an appropriate algorithm for slow variation data series measurements. We can notice that EEE^* does not provide greatest estimation quality for the radioactivity, as shown in Figure 3.6(c). However, we note that $\sim$ 90% of relative errors produced for the radioactivity were around 8%. In what follows, we try to observe the correlation between real data and predicted ones. The purpose of this is to see how much the real data and the samples produced by the discussed models are correlated. In other word, if two variables are correlated, we can predict one based on the other.

3.10.2 RV coefficient

We refer to the correlation coefficient to observe the relationship between values. We can notice in Table 3.5 that the correlation coefficient tends to 1 in EEE^*. It is greater than other models correlation coefficient values. This indicates a strong linear relationship between the value produced by our model and the real data. According to these results and the estimation errors presented in Table 3.4, we consider that our model is a good prediction model. It produces more accurate estimations then AR/ARMA models.

Data type	AR(2)	AR(3)	ARMA(3,3)	EEE^*
Garden (T°)	up to 43%	up to 50%	Divergence	up to 7%
Chemical process (T°)	up to 40%	up to 43%	up to 41%	up to 6%
Carbon dioxide	up to 10%	up to 11%	up to 11%	up to 2%
Cardiac frequency	up to 50%	up to 50%	up to 35%	99% of errors were $\leq$ 5%
Radiosity	up to 60%	up to 60%	up to 60%	99% of errors were $\leq$ 8%
Female (T°)	~5%	~5%	~4%	~3%

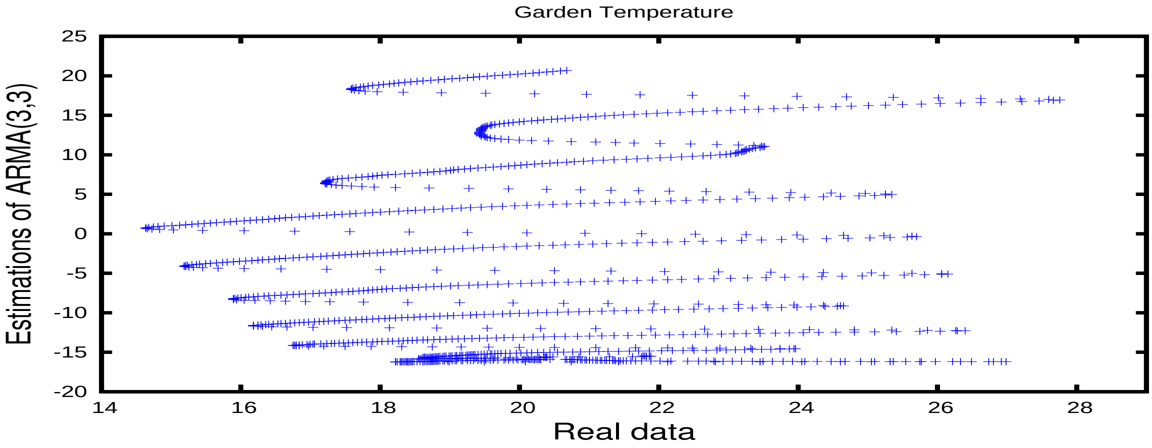
Table 3.4: The relative error produced by each model estimations: AR(2),AR(3), ARMA(3,3) and EEE^*

3.10.3 Scatterplots

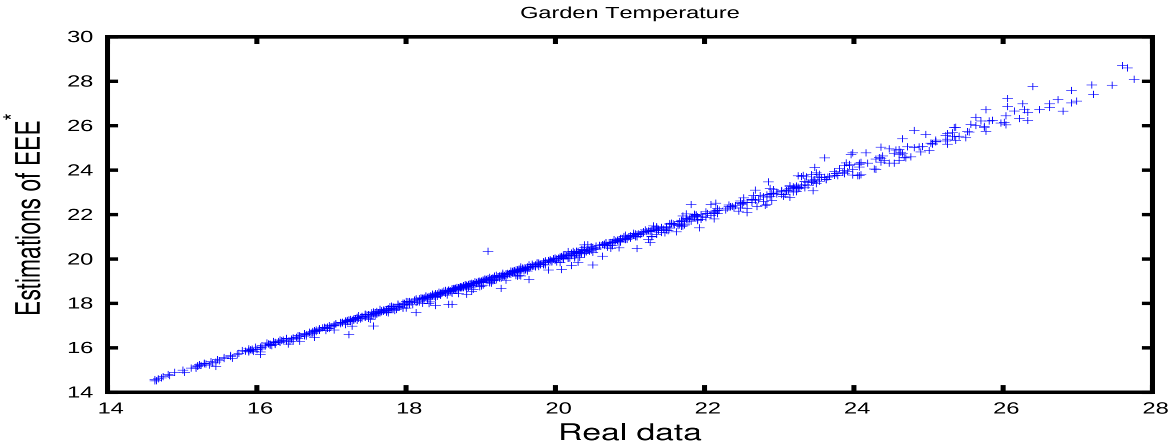
Having a correlation coefficient value that tends to zero (such as for AR/ARMA in Table 3.5), does not mean the absence of relation. There may be a non linear relation between the models outputs and the real data. The possibility of such non-linear relationships is another reason why examining *scatterplots* is a necessary step in evaluating every correlation.

	Garden Temperature	Chemical process (T°)	Carbon dioxide	Cardiac Freq.
AR(2)	~1.956 e-08	~1.586e-06	5.58e-06	3.343e-06
AR(3)	~0.01523	~3.363e-05	6.90e-07	2.752e-08
ARMA(3,3)	0.00439	0.018	0.00012	0.00012
EEE^*	0.996	0.9889	0.983	0.964

Table 3.5: RV value measuring the correlation between real and predicted values by AR(2), AR(3), ARMA(3,3) and EEE^*



(a) *Garden*: $ARMA(3,3)$



(b) *Garden*: EEE^*

Figure 3.8: Scatterplots for some data types.

Figure 3.8 shows the relationships between the ARMA(3,3) and EEE^* models and

$CO2$ data values. It indicates that EEE^{*} is a suitable algorithm to predict data studied here then ARMA(3,3) model. As the scatterplots for other models revel similar results using the different data types studied, we did not show them.

3.11 Conclusion

We aim for a long-lived sensor nodes by effectively reducing their energy consumption. More specifically, we are interested in energy consumed for transmission. We adopt a cluster-based data aggregation scheme. We focus on reducing the transmission cost between the source node and its sink (i.e. the Cluster-Head). Based on AR model, we proposed two forecasting algorithms (EEE and EEE^{+}). We evaluated their efficiency for transmission cost and accuracy. The experimentation results showed that EEE^{+} algorithm is more adaptive. It shows an efficiency of about 70% reduction in transmission overhead. It also produces good predictions accuracy based on the experimentations we have made. The relative errors for predictions were $\sim$5%.

In the other hand, we propose a method to calculate a dynamic prediction bound. The motivation behind that is to observe its impact on accuracy. In fact, a threshold is a constraint for data calibration along the process of prediction. The selection of a suitable bound can avoid model adjustment and of course the need for communication. For this purpose, we have extended EEE^{+} to EEE^{*} with a dynamic threshold calculation. The algorithm relies on the data distribution over time to estimate the prediction bound. Note that we consider true values the data used for experimentations. We also assume that they do not contain erroneous or anomalous values. Abnormal behaviors detection in data stream will be discussed later in Chapter 5.

The experiments on different data types show that the relative errors for prediction do not exceed 8% for most used data types. However, The errors produced by other models were higher. Therefore, using our proposed model for data prediction, it is possible to reduce the rate of model adjustment and communication overhead. Thus, it is possible to decrease the number of computational operations. Hence, the node's lifetime may be increased which of course impacts the network longevity.

This work is part of an incremental process of analysis, experimentations and studies to be done to reach energy-efficient algorithm. In the next chapter, we will investigate data similarity measurements based on the spatial correlation. We propose a data similarity detection algorithm to observe similar data in order to reduce the transmission overhead.

Basic research is like shooting an arrow into the air and, where it lands, painting a target.
$\sim\sim\sim\sim$*Homer Adkins*

Chapter 4

Data Similarity in In-network processing

Summary

Due to the dense deployment in WSNs, neighboring nodes that sense the same property and go through the same environmental conditions, are expected to provide similar information. Thus, data communicated between nodes may be duplicated or redundant. This may have its negative impact on the transmission cost for the limited energy resource sensors. For this purpose, we present an algorithm to detect data similarity during aggregation. The main goal, is not only decreasing the amount of transmitted data towards destination. It is also to enable the destination to accurately reconstruct the information, even with the fewer data amount it receives. The algorithm is based on the notion of prediction proposed in Chapter 3. Experimentation results show a possibility to reduce data transmission and provide an accurate data reconstruction.

Contents

4.1 Introduction

Nodes in WSNs gather and rely information to one or more other nodes. One node, called the leader, collects data from surrounding nodes and then sends the summarized information to upstream nodes. With the many to one nature of WSNs, the dense deployment of sensors and the use of multi-path routing [165, 172], data may be duplicated. It may also happen that some nodes notify the sink about the same event, at almost the same time, and approximately the same values. This induces a propagation of redundant highly correlated data. Redundancy is costly in terms of system performance. It results in energy depletion, network overloading and congestion. Such data correlation can be related to the environmental conditions that nodes go through, and to their spatial distribution. Tobler's first law of geography states that "Everything is related to everything else, but near things are more related than distant things" [231]. This statistical observation implies that data correlation increases when the spatial separation decreases. For example, when the sun shines through a room's window, nodes near the window tend to be hot, the ones nearby are a little cooler, while the temperature of nodes on the other end of the room depends on variations across the room.

Our goal is to observe similarity among collected data in order to reduce the transmission overhead. In this chapter, we rely on the temporal data correlation, discussed in Chapter 3. We employ the spatial data correlation to enhance our work and decrease the transmission cost. Exploring data correlation (temporal/spatial) in the sensor readings induces opportunities for data aggregation, which are very important considering severe energy constraints of the sensor nodes.

In this chapter, data aggregation is performed in a cluster based manner. Nodes communicate their data to their cluster head. The latter will calculate the similarity amongst the values. It checks the spatial correlation among sensors readings to observe redundancy and reduce the transmission overhead. However, redundancy is both ally and enemy [59]. Collected redundancy can be used to improve the reliability of the entire system. For this purpose, we decide to communicate the similarity degree among the redundant data. Thus, we keep redundancy during transmission without harvesting the source of energy. The tradeoffs between sending less amount of data, and the possibility to accurately reconstruct the original data on the sink, were among the challenges in this work. Our algorithm shows its ability to reduce the total transmission overhead in terms of bits. It reduces the transmission overhead by $\sim$ 10% to 20% depending on the data type used in the experimentation. Nonetheless, it shows an accurate reconstruction of the original data on the sink. Experimentation results show that the relative errors between estimations made by the source sensors and the ones deduced by the sink, are very small. They belong to $[-7 \times 10^{-7}, 10^{-6}]$ (see Section 5.6).

This chapter is structured as follows: In Section 4.2, we introduce the notion of similarity and present the similarity function used in this work. To measure the data similarity we rely on kernel-based methods, due to their popularity in different applications [198]. We also observe the factors behind data correlation and the type of redundancy in WSNs. Section 4.3 provides an overview of some recent spatio-temporal techniques in the domain. We present our similarity detection algorithm in Section 4.4. Our work takes advantage of spatio-temporal data correlations to detect and reduce redundancy. The main goal is to reduce the communication cost and energy consumption. In Section 4.5, we observe the quality of data reconstructed at the sink, as well as teh transmission overhead. Section 4.6, concludes this chapter.

4.2 Preliminaries

4.2.1 Notion of similarity

Similarity measures play a central role in reasoning in many applications [4, 47, 103, 134] such as bioinformatics, natural language processing (NLP), image processing, pattern recognition and different other problems of information retrieval. The similarity functions, called also affinity functions, are denoted by $s : X \times X \longrightarrow R$. They are in some sense the converse to dissimilarity functions. This means that the similarity between two objects should grow if their dissimilarity decreases. In particular, a similarity function is supposed to increase the more similar the points are [145]. In what follows, we present a brief overview on special type of similarity functions: the "Kernel functions". We will use this type of function later in our algorithm (due to its popularity [198]). The goal is to detect similarity among aggregated data. In what follows, we provide an overview on kernel-based methods. We also introduce the kernel function used in this chapter.

4.2.2 Kernel-based methods

In the last years, a number of powerful kernel-based learning machines have been proposed. Among these we mention: Support Vector Machines (SVMs) [25, 195, 197, 234], kernel Fisher discriminant (KFD) [156, 157], and kernel principal component analysis (KPCA) [158, 196]. Successful applications of kernel-based algorithms have been reported for various fields: Optical pattern and object recognition [22, 65], text categorization [76, 125], time-series prediction [109, 162], gene expression profile analysis [31, 89], DNA and protein analysis [119, 274], and many more[1]. Kernel functions are one of the most popular tools in Machine Learning. They have been used for relation extraction within the field of information extraction, such as, in text recognition, images analysis, ranking items, etc. They were used to discover if two or more candidate entities are related and which relation includes between them.

Kernel-based learning algorithms [56] work by embedding the data points into a Hilbert space, and searching for linear relations in such a space. The embedding is performed implicitly, by specifying the inner product between each pair of points rather than by giving their coordinates explicitly. This approach has several advantages. The most important one derives from the fact that often the inner product, in the embedding space, can be computed much more easily than the coordinates of the points themselves.

Suppose having an input set X, an embedding space F and a map $\phi : X \rightarrow F$. For two points, $x_i \in X$ and $x_j \in X$, the function that returns the inner product between their images, in the space F, is known as the $kernel function$.

Definition 4.2.1. *: A kernel k is a function, such that k(x, z) = < ϕ (x), ϕ (z)> for all x, z $\in X$, where ϕ is a mapping from X to an (inner product) feature space F.*

[1] See also Guyon's web page http://www.clopinet.com/isabelle/Projects/SVM/applist.html on applications of SVMs.

Kernel-based methods rely on measures of similarity called kernel functions. They perform several tasks for pattern recognition [87, 198, 204], such as classification, regression. They find and study different types of relations in many types of data. Kernels can be general or tailored to the specific domain from which the data arises. Among the generic kernels, we mention the popular gaussian kernels (4.1) [198, 204]. These kernels are widely used in many domains. For exmaples, they are used in statistics, where they describe the normal distributions, in signal processing where they serve to define Gaussian filters, in image processing where two-dimensional Gaussians are used for Gaussian blurs, etc. We also mention other specific kernels, such as the Fisher kernels[120] and kernels for specific structures, like data strings and tree [237]. Such kernels can be used for biological sequences samples where data is represented as strings [77], and Natural Language Processing (NLP) where the data is in a form of parse tree [49].

We use the following gaussian kernel function due to its popularity:

$$k(x, y) = \exp \frac{- \parallel x - y \parallel^2}{2 * \sigma^2} \tag{4.1}$$

where $k(x, y) \in [0, 1]$, σ determines the width of the Gaussian kernel (Note that we have $x = y$ when $k(x, y) = 1$). We initialize $\sigma = 1.74$ as in [200].

This section was a preliminary for the proposed algorithm in Section 4.4. We highlighted the importance of kernel-based methods and introduced the kernel function used in this work. Next, we observe the similarity on nodes and their impact on the network.

4.2.3 Similar data and redundancy in WSNs

Correlation in sensors' data occurs due to the similarity of environmental factors for sensors. Nodes can have similar environmental conditions and spatial factors, which normally produce similar or exact data. The communication of such data between nodes, yields for redundancy which increases the energy consumption of the network. Nonetheless, the dense deployment of sensor nodes to achieve satisfactory coverage [7] as well as the mutli-path data routing (i.e broadcast), can increase the amount of redundancy. Thus, information about an event can be captured by many surrounding sensor nodes. This generates a large amount of traffic and consumes a lot of battery energy. However, if redundancy is detected and reduced, unnecessary energy consumption can be omitted.

Redundancy among data in Wireless Sensor Networks can be spatial or temporal [59]. The spatial redundancy occurs if spatial data correlation exists. This means, when observations from the nodes that are in close proximity, are highly correlated. Furthermore, the nature of the physical phenomenon constitutes the temporal correlation between each consecutive observation of a sensor node. This is the temporal correlation. These spatio-temporal correlations, and, the collaborative nature of WSNs, bring significant potential advantages for an efficient data communication. Most existing approaches try to organize sensors into groups based on their spatial relationships [46, 102, 169, 170, 203, 252]. This induces opportunities for an efficient data aggregation in WSNs, especially energy-limited sensor nodes. In what follows, we provide an overview of some recent spatio-temporal techniques in the domain. Then, we present our similarity detection algorithm and show its efficiency.

4.3 Related works

A significant challenge in WSNs is to prolong the monitoring operation of sensor nodes by efficiently using their limited energy, bandwidth and computation resources. By allowing the nodes to corporate to carry out joint data from aggregation, the amount of data communicated within the network can be reduced. Recent techniques for processing multiple sensor streams in an energy efficient manner has been proposed. These techniques make use of both spatial and temporal correlations, as well as, clustering approaches to perform data reduction. In this section, we present some of these techniques.

The work in [106], proposes an algorithm that manages spatio-temporal data in a distributed fashion. It performs in-network regression using kernel functions assuming rectangular regions of support. The network is assumed to contain multiple overlapping regions. In each region significant spatial correlations are expected to be observed. Authors introduce the usage of kernel functions. They model the event, within each region, using some basis functions. Data is approximated by a weighted linear combination of the basis functions. They model spatial correlation using linear regression model of weighted kernel functions.

Other technique, such as [68], proposes the use of statistical models of real world processes. Its goal is to reduce the cost of sensing and communication in sensor networks. The built prototype is Barbie-Q (BBQ). It consists of a declarative query processor and a probabilistic model and planner, based on time-varying multivariate Gaussians. The model runs at the base station, where users submit SQL-like queries that include error intervals and confidence bounds. Another prototype built is Ken [48]. Ken utilizes a pair of dynamic probabilistic models. One runs at the base station (sink) and the other runs at each node in the network. It proposes a framework for selecting approximate data with bounded-loss guarantees. The aim is to minimize the costs of communication throughout the network. Ken bears several similarities to BBQ, as it evaluates a probability density function and exploits both spatial and temporal correlations between sensors. Their key difference lies in the fact that BBQ is pull-based, i.e. the base station acquires data from the network only when necessary in order to satisfy queries. While Ken is push-based. This means that source nodes monitor their data continuously. They only route values towards the sink when its model needs to be synchronized with the local ones. In effect, Ken addresses outliers detection, the basic weakness of BBQ. In order to utilize spatial correlations, Ken partitions the sources into clusters, called cliques. The inference is done on one node per clique, the clique root.

Our algorithm extends a previous work [99, 100] (already described in Chapter 3) to integrate spatial similarity measurements. The main goal is to reduce the transmission overhead in WSNs, in order to increase nodes' lifetime. In the previous works, we adopted a cluster-based data aggregation scheme. We focused on reducing the transmission rate and overhead between the source node and its sink (i.e. the Cluster-Head). We proposed an estimation algorithm that resides on time series forecasting described in Chapter 3. A Sensor and a sink use the same prediction model. The communication between them is restricted to a threshold violation. During communication, a sensor S sends, instead of previous data raw samples to the sink, certain previous error values e_i^S. The aim is to re-learn the prediction model at the sink, and improve its performance. Our experiments show that the transmission cost can be reduced (about 70%) (see Chapter 3). On the other hand, the prediction accuracy has been also observed. Results show a relative prediction errors between 2% and 8% depending on the data type. In this chapter, a cluster-head (i.e. the aggregator) checks for spatial correlation among the data it collects, on order to reduce the amount of data transmission.

Due to the spatial correlation in the sensed data, aggregation techniques were incorporated into the routing protocols. Different routing strategies involve data compression via coding in correlated data aggregation, to reduce data traffic. These strategies (such as aggregation with Distributed source coding strategy-DSC [55]) have been based on lossless coding schemes, such as Slepian-wolf coding in data aggregation [53]. In this work, we do not propose a clustering/routing methods or discuss a coding schema and its dependence on optimal cluster-sizes. We suppose that clusters and clusters heads (CH) are determined using distributed or centralized methods such as LEACH, HEED [114, 264] or others [206]. We assume having a suitable routing protocols that do not interfere with the spatial aggregation described in Section 4.4.

In our work, each non cluster head node sends data to the CH in its own cluster instead of the base station (BS).
The approach of clustering has the following advantages:
1) Non-CH nodes can save energy consumption. They have to send data to their own CH nearby. Thereby, they can avoid long-distance communication.
2) The amount of data sent towards the base station (BS) can be reduced due to data fusion, which again saves energy.
Our work relies on the idea of decreasing the communication overhead between nodes by:
1- Detecting and reducing redundancy using data similarity measurements.
2- Reducing the number of communicated bits, since it was indicated that the transmission of one bit over 100 meters would cost about the same level of energy as executing 3000 instructions in sensor node [92, 230]. In what follows, we present the proposed algorithm for data similarity measurements.

4.4 Our data similarity detection algorithm

Given a typical WSNs in which each node records information from its vicinity and transfers this information to a centralized base station. Nodes which are close to each other, eventually sense similar information. Hence the information is geographically correlated. So, before sending it to the central agent, a huge saving in transmission cost may be achieved. That is by aggregating information from nearby nodes, removing redundancy and keeping data transmission to a minimum. We will call these geographical regions: "similarity regions", denoted by R_j. We consider that these regions are pre-defined on the base station at the deployment time. Clusters are determined based on nodes' battery level, their coverage capabilities, the communication cost and the node density, such as in [35, 114, 264]. So, it may happen that two neighboring clusters share spatial data correlation. Hence, a similarity region may contain different clusters spatially correlated as shown in Figure 4.1. We denote by Λ_j the number of clusters in a similarity region R_j.

Our goal is to detect data similarities during aggregation to keep data transmission overhead to a minimum. During aggregation (intra-/inter-clustering aggregation), an aggregator verifies the source of each received data. It checks if it is a source node inside its own cluster or a CH node of a neighboring clusters. It also examines the degree of similarity among the received data. Note that along this process of similarity detection, abnormal behaviors may be observed. However, detecting such behaviors are out of the scope of this chapter, and will be observed later in Chapter 5. Returning to our algorithm, let us first describe some notations used in this section:

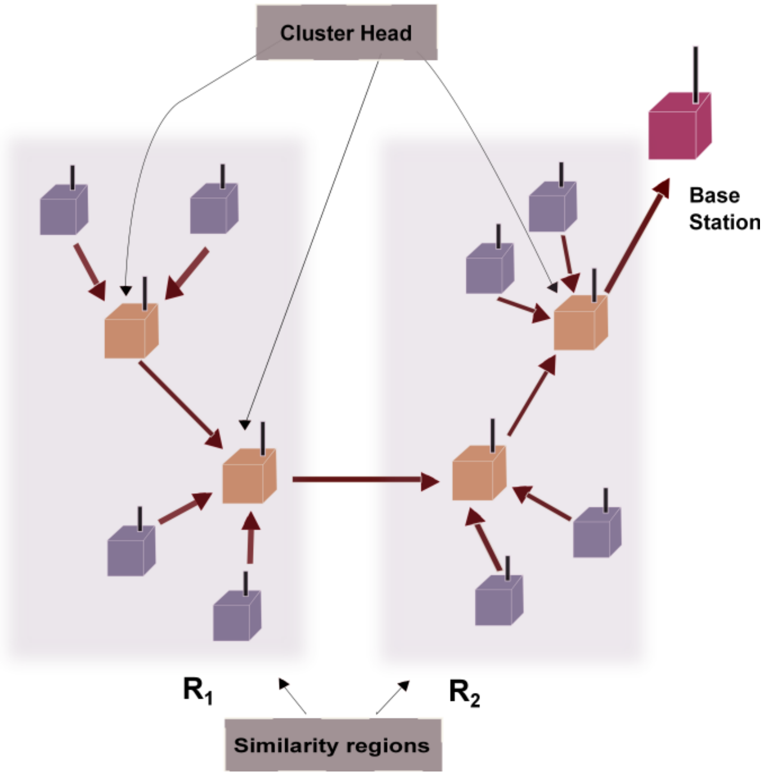


Figure 4.1: The concept of similarity region.

- δ: the spatial similarity degree threshold. More specifically, δ_{R_j} and δ_c are respectively the degree of similarity thresholds inside a similarity region R_j and cluster. They are set during the clustering and the definition of similarity regions.

- Γ: the data similarity threshold. Precisely, Γ_{R_j} and Γ_c refers respectively to the degree of data similarity inside a region R_j and a cluster. The choice of Γ values could be a user defined threshold when clusters and similarity regions are determined.

- $d_{n_1 n_2}$: the distance between two nodes n_1 and n_2. For simplicity, we refer to the euclidean distance.
  For every pair of nodes n_1 and n_2 belonging to the same cluster, $d_{n_1 n_2} \leq \delta_c$. While belonging to the same similarity region, $d_{n_1 n_2} \leq \delta_{R_j}$. We denote by d_{SCH} the distance between a source S and a cluster head CH. If $\Lambda_j = 1$, the similarity region R_j is a cluster with $\delta_{R_j} = \delta_c$ and $\Gamma_{R_j} = \Gamma_c$.

- We choose to capture the (dis)similarity between two arrays of data A and B of length p by $K(A, B) = \Pi_{i=0}^{p} k(A_i, B_i)$.

In this work, the temporal data correlation is modeled using EEE^* algorithm [100] (see Chapter 3). Each sensor S uses a prediction model to estimate future environment values. It communicates with the sink only when a prediction threshold violation occurs. The values to be transmitted are no more the model parameters or recent data raw, but a recent number p of error values e_i^S where $i \in \{0, .., p\}$. An aggregator, while monitoring the environment, may receive an array of data error values (e_i^S) from a source sensor in its cluster or a CH node in the neighborhood. It may combine these values with its own error values e_i^{agg}, if it has, then routes them to the base station. However, during aggregation it is important to distinguish between the fused data. Simply speaking, fusing similar data can ensure redundancy and lead to huge communication cost. For this purpose, we try to observe the spatial correlation. For two nodes n_1 and n_2 that are located in the same similarity region or cluster, the transmission overhead may be reduced, if their values $(e_i^{n_1})$ and ($e_i^{n_2}$) are highly correlated. The benefit of similarity detection is not only restricted to redundancy detection. It may ensure reliable abnormal behavior detection during inter-clustering fusion. For example, suppose that a CH node sends a value to another CH resided in the same similarity region. If the values are not similar, while they should be, one can deduce that something interesting happened.

We assume that neighbor nodes monitor the same event, the position of each sensor is predetermined. Our algorithm is presented as follows:
A sensor node S sends data array (e_i^S) to a CH node (based on EEE^* algorithm). The latter -before starting tranmission- calculates the spatial degree of similarity between its own values and the aggregated ones. This is to ensure if they are (i.e the CH and source nodes) in the same cluster or similarity region.

- If $d_{SCH} \leq \delta_c$: The source and the CH nodes are in the same cluster. Then, the aggregator uses the Gaussian kernel function to calculate the degree of data similarity $K(e^S, e^{CH}) = \Pi_{i=0}^{p} k(e_i^S, e_i^{CH})$ between its array of data (e_i^{CH}) and (e_i^S) of the source.
    - if $K(e^S, e^{CH}) \geq \Gamma_c$, the values are highly spatially correlated and redundancy occurs. Then the aggregator routes its own data values (e_i^{CH}) toward the sink. However, the sink needs to accurately update its prediction model to improve the quality of estimation. Therefore, CH sends in addition to its own values, the array of similarity measures $k(e_i^S, e_i^{CH})$ between its data values and the sensors' ones. These values belong to $[0, 1]$. We consider that sending their decimal part (integer values) instead of the main error values (e_i^S) (float values) can help in reducing the data traffic in terms of number of bits.
    - if $K(e^{CH}, e^S) \prec \Gamma_c$. This indicates that an anomaly occured, sensors are misbehaving or that something interesting has happened. For example, the sensor became hot due to a fire that started nearby. The sink should be aware of such abnormal cases. Thus, the aggregator decides to send both values e_i^{CH} and e_i^S toward the sink.
- Otherwise, if sensors are in the same similarity region $(d_{SCH} \leq \delta_{R_j})$, the CH node follows the same process mentioned above by changing Γ_c to Γ_{R_j}

- Note that if both sensors are not in the same similarity region, the aggregator decides to send both values e_i^{CH} and e_i^S.

In the following section, we assume that there is no data loss. We propose having a suitable routing methods and coding scheme. We apply our methodology on a simple case study. We focus on observing the transmission overhead and the data prediction accuracy between the sink and the source sensors. Note that a sensor and a sink use the same time series prediction model (with $p = 3$). The degree of similarity between two values is set according to an application requirements. In the experimentation below, we considered that two values are similar when their degree of similarity is greater then 30%. Hence, $k(e_i^S, e_i^{CH}) \geq 0.3$. Since $K(e^S, e^{CH}) = \Pi_{i=0}^{p} k(e_i^S, e_i^{CH})$. Then, $K(e^S, e^{CH}) \geq (0.3)^p$. We then set $\Gamma_c = 0.027$.

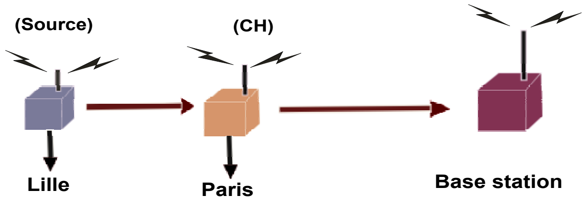


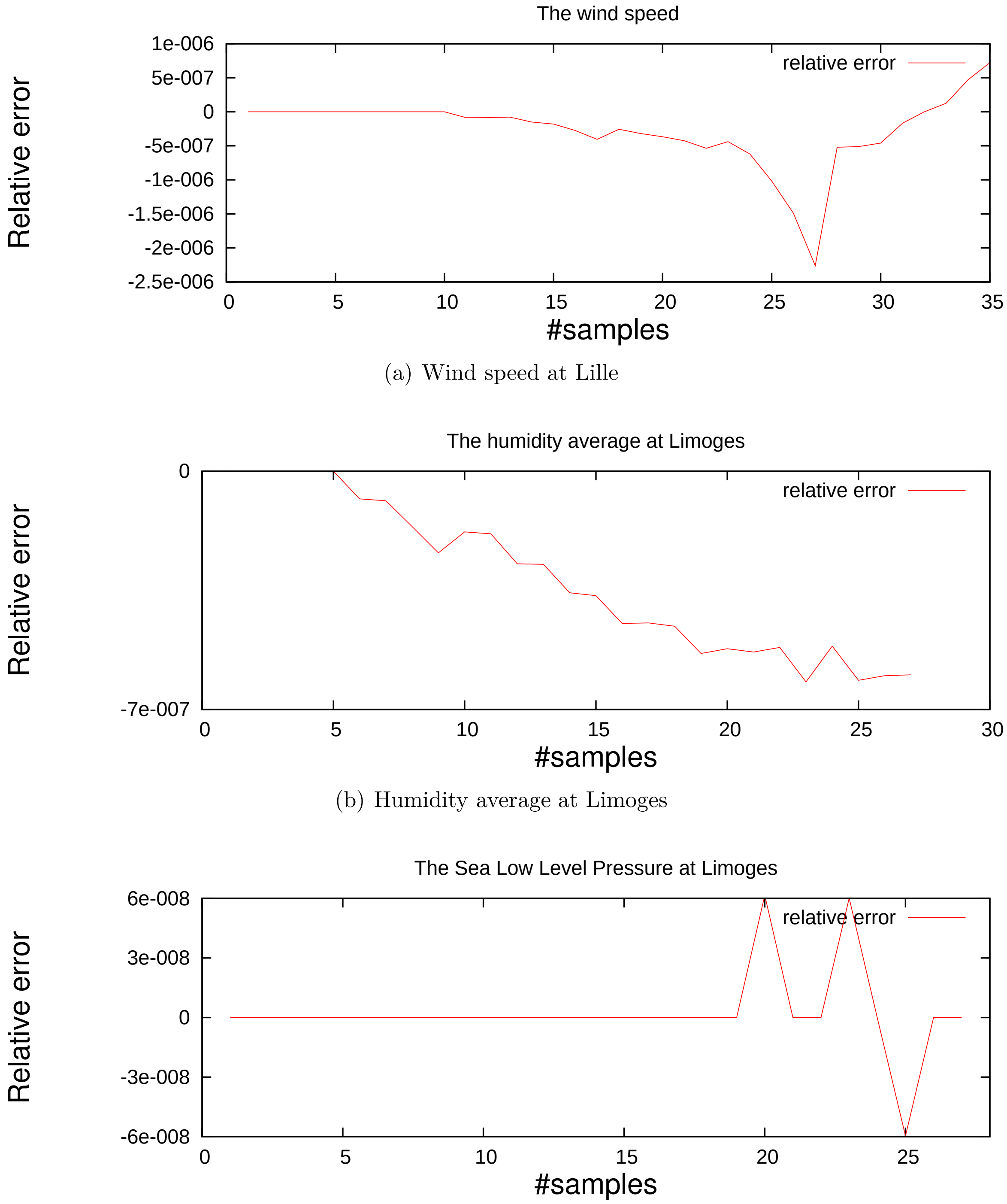
Figure 4.2: Wind speed measurement: the source sensor is located at Lille while the cluster head at Paris.

4.5 Experimentation and accuracy results

We applied our algorithm on a simple 2-hop network topology composed of a sensor, an aggregator and a sink. We assume that the source sensor and the CH nodes are in the same similarity region R_j. Thus $\Lambda_j = 1$, $\Gamma_{R_j} = \Gamma_c = 0.027$. Since our prediction model in [99, 100] is suitable for slow variation measurements, we applied our algorithm on such real data values[2] : Wind speed at Lille (aggregator at Paris) (see Figure 4.2), Humidity average at Limoges and Sea Low Level pressure at Limoges (aggregator at Lyon). Currently, we observe the impact of similarity detection on the transmission cost reduction. We also observe the quality of prediction at the sink, since it updates its model according to the received values.

Figure 4.3 shows that the relative error values, between source and sink estimations, belong to $[-7 \times 10^{-7}, 10^{-6}]$. This means that data are acuratlyreconstructed at the sink. In addition, Table 4.1 shows the number of data aggregated and communicated to the base station, before and after using the similarity measurement. The data could be the error values (float numbers) and/or the similarity degree (integer values) according to the similarity algorithm. We can see that the number of floats communicated using the similarity measurement is reduced about 41% for wind speed, $\sim$ 20% for humidity and $\sim$ 39% for sea low level pressure. This indicates that energy saving can be increased, since the number of transmitted bits is reduced. Table 4.2 represents the total overhead in terms of bytes, before and after using the similarity measurements. If we consider that

[2] http://www.wunderground.com/global/stations/



(a) Wind speed at Lille

(b) Humidity average at Limoges

(c) Sea Low Level Pressure at Limoges

Figure 4.3: The relative error between the sensor and sink estimations.

an integer is represented on 4 bytes and a float on 8 bytes. Our experiment shows a reduction, in terms of bits, of about ∼ 20% for wind speed, ∼ 10% for humidity and ∼ 20% for sea low level pressure.

	Using sim. meas.		**Without sim. meas.**	
Data traffic	*# float(a)*	*# int.*	*# float(b)*	$\frac{(a)}{(b)}$
Wind speed	51	36	87	~ 0.586
Humidity	24	6	30	0.8
Pressure	69	42	111	~ 0.62

Table 4.1: The data traffic before and after using the similarity measurements. Here $\frac{(a)}{(b)}$, is the fraction between the data traffic (in terms of floats) before and after using similarity measurements .

Data traffic	***Without sim.(a)***	***Using sim.(b)***	$\frac{(b)}{(a)}$
Wind speed	696 bytes	552 bytes	~ 0.79
Humidity	240 bytes	216 bytes	0.9
Pressure	888 bytes	720 bytes	~ 0.81

Table 4.2: The data traffic in bytes before and after using the similarity measurements.

4.6 Conclusion

The key idea in this chapter is to observe data correlation during aggregation. We consider a cluster-based scheme. The similarity detection among data is performed on the intermediate nodes (i.e. the aggregator nodes). In our work, all nodes are capable to make predictions using the EEE^* algorithm proposed in Chapter 3. Nodes communicate their values to the aggregator, which in turn, checks out for similarity amongst the received ones. The presence of spatial correlation is checked before transmitting data to the destination. By allowing the nodes to corporate to carry out joint data from aggregation and check for similarity, the amount of data communicated within the network can be reduced. However, collecting and transmitting less amount of data was not the only concern in this chapter. Another concern was the significance of the transmitted data and their ability to reconstruct the original data structure at the sink. Since the transmission is restricted to a prediction bound violation, the sink node may ask for new values to update its prediction model. These values are aggregated from the sensor nodes. During this aggregation, less amount of data is expected to be collected for the sake of energy saving. Hence, it is important to observe the quality of collected data, to provide the sink node with accurate values so it can continue predicting correctly. Our experiments show that it's possible to reduce the communication overhead between nodes while ensuring a reasonable data quality and accuracy. We noticed that the total overhead can be reduced about 20% for wind speed, 10% for humidity and 20% for sea low level pressure. As for the quality of predictions between a source sensor and a sink, the relative errors were in $[-7 \times 10^{-7}, 10^{-6}]$. Thus, by taking advantage of the spatial correlation and introducing the similarity measurement, significant advantages are gained in terms of energy savings and accuracy. In our future work, we aim at enhancing our algorithm. Since our experiments adopt a simple case topology. We will continue to observe the performance of our algorithm on complex topologies with clustering and routing methods. The following chapter tackles abnormal behaviors detection in sensor nodes.

To study the abnormal is the best way of understanding the normal.
∼∼∼∼*William James*

Chapter 5

Anomaly Detection in WSNs

Summary

Data analysis in Wireless Sensor Networks faces different challenges. One of the main challenges is to identify normal and abnormal (anomalous) behaviors with high accuracy. The importance of anomaly detection is due to the fact that anomalous data can significantly impact on the behavior of an application in a variety of domains. If such data is not detected, inaccurate information can be produced. Such information can change the behavior of an application to a critical and unwanted actions. For constrained WSNs, additional challenge subsists which is the energy consumption. Sensors have limited resource capabilities. Moreover, they need to examine data coming from many streams, dynamically, to avoid combining untrusted ones or make unnecessary actions/communication. In this chapter, we propose an algorithm for temporal anomaly detection in Wireless Sensor Networks. Our experiments results show that the proposed algorithm can accurately detect abnormal behaviors in sensor measurements. It also produces low false alarm rate for slow variation time series measurements without harvesting the source of energy.

Contents

5.1 Introduction

Abnormal data detection is a long studied problem in data analysis. It has various names: outliers detection, novelty detection, fraud detection, anomaly detection, etc. The term abnormal data detection seeks to identify instances of unusual activities present in the data [37, 269]. Or in another words, it is the identification of measurements that deviate markedly from the normal pattern of data [37] in many application domains. For example, speech or handwritten digit recognition [58], radar target detection [13], novelty detection in robot behavior [57], detection of malignant tumors in mammograms [213, 225], etc.

Detecting abnormal behavior in sensor networks is an interesting issue. It is important to keep observing reliability of sensor data in order to make appropriate decisions and extracting useful knowledge. The presence of inconsistent data may cause undesirable outcomes. An unreliable mobile node produces unpredictable motion and behaviors which affect its source of energy and deliver fake information or knowledge. Malicious nodes may divert traffic from the base station. They may modify the collected data to cause the base station to make mis-informed decisions, etc. WSNs are susceptible to many types of attacks because they are deployed in open and unprotected environments. They incline to outliers due to several factors, such as malicious attacks, changes in the observed external phenomena or errors in communication, etc [269].

Research on the detection of outliers/anomalies in WSNs include: fault detection [144] (such as noise, errors), event detection [132, 151] (such as fire, explosion detection) and intrusion detection [19, 62] such as malicious attacks. These techniques are widely employed to remove inaccurate data in order to improve data quality. However classifying a behavior as abnormal is not trivial. It is related to define a normal region that encompasses every possible normal behavior, which is often difficult [272]. In fact, most data in WSNs are coming from unknown generative processes. In other words, the real data distribution is not known, whether in environmental monitoring (temperature, humidity, etc) or in medicine where patients respond differently to treatments, etc. Thus, the idea of abnormality detection techniques is to build a model of normality of the training data using only normal examples and then compare the test patterns against this model. They try to approximate the distribution of the data and then, they term outliers the values that deviate significantly from that model [19, 57, 81, 149, 188, 190, 201, 225, 233, 269].

In our work, we detect data anomalies with respect to previous history of collected data over time. We first model time series data to predict behaviors. Then, we define some ranking conditions under which a reading is marked as abnormal. In our detection techniques, a sensor marks a reading as outlier depending on the extent to which the reading disagrees with the data distribution history. In other words, when the reading breaks the ranking conditions. These ranking conditions are calculated from the history of normal data. The main challenge in our work is to accurately model the data and detecting abnormality. The goal is to achieve this with low false alarm rate and less energy consumption. Note that a false alarm refers to a value ranked abnormal while it is normal. We use the prediction model proposed in Chapter 3 (EEE^* alogorithm). We adapt dynamically the model of data distribution and the ranking conditions to the changing environmental conditions (i.e. to the temporal data correlation). In this work, we oriented our efforts to improve data quality related to the accuracy of anomaly detection not only the quality of prediction. Our experimentation results show that our proposed technique can detect about 78% up to 90% of slipped anomalies. The undetected ones were the values that are too close to the normal ones. They were generated by a perturbation rate less than 1% on the real values. Nonetheless, results show the efficiency of our technique concerning its low false alarm rate (∼8%).

This chapter is organized as follows: Section 5.2 provides an overview of previous studies in the fields of outlier/event detection. Section 5.3 introduces the motivation and contribution our work. Section 5.4 presents the different notations, equations and terminologies we will use throughout the chapter to describe our detection technique. In Section 5.5, our outlier detection technique and its expected benchmarks are presented. Section 5.6 discusses the experimentation results and Section 5.7 concludes this chapter.

5.2 Related works

Many outlier detection techniques have been developed for WSNs, some are: *statistical-based*, *nearest neighbor-based*, *clustering-based*, *classification-based techniques*.

Statistical-based approaches are among the earlier and widely used anomaly detection techniques. Some of these approaches are Gaussian-based and kernel-based approaches. In [255], two local Gaussian-based techniques were presented for identification of outlying sensors as well as identification of event boundary in sensor networks. These techniques employ the existing spatial correlation among the readings of neighboring nodes to distinguish between outlying sensors and event boundary. Accuracy of these outliers detection techniques is not relatively high due to the fact that they ignore the temporal correlation of sensor readings. In ecological applications of WSNs, [18] identifies errors and detect events using local technique based on spatio-temporal correlations of sensor data. The detected anomalous measurement may be considered as event if it is likely to be temporally different from its previous measurements but spatially correlated. Its main drawback is that it relies on the choice of the appropriate threshold values.

A kernel-based technique for online identification of outliers in streaming sensor data have been proposed in [167]. It uses kernel density estimator to approximate the underlying distribution of sensor data. Thus, each node can locally identify outliers if the values deviate significantly from the model of approximated data distribution. Another work [233] is called PAQ (for Probabilistic Adaptable Query system). It exploits temporal/spatial correlation using a combination of AR models (AutoRegressive models) to probabilistically answer queries. PAQ allows the sensor to detect data anomalies with respect to previous history, where a data anomaly is a sensor value that the model does not predict to within the user-specified error bound.

In *nearest neighbor-based* (distance-based) approaches, anomaly detection is based on the average distance between every data measurements to its corresponding q^{th} closest neighbor. If the distance from a given data measurement to its q^{th} closest neighbor is significantly bigger than the average, then the data measurement is considered as an outlier [223]. The work in [27] proposes a technique based on distance similarity to identify global outliers. Each node uses distance similarity to locally identify outliers and then broadcasts the outliers to neighboring nodes for verification. The technique does not adopt any network structure so that every node uses broadcast to communicate with other nodes in the network, which will cause too much communication overhead.

Clustering-based techniques seek to divide a data set into several groups. Data points belonging to the same group are more similar to one another than they are to those in a different group. Data are considered as outliers if they do not belong to clusters or if their clusters are significantly smaller than other clusters. Euclidean distance is often used as the dissimilarity measure between two data instances.

Classification-based approaches are categorized as Bayesian network-based and support vector machine-based approaches based on type of classification model that they use.
The main challenge for all outliers detection techniques for WSNs relies on the accuracy requirements and the maintaining of less resource consumptions [79]. The reliability and accuracy of sensor data before the decision-making process is extremely important.
Our model is designed with the energy restrictions in mind. It aims at monitoring changes in the physical phenomena, detecting outliers and reducing the communication between a sensors and its sink. In this work, we use a statistical technique combined with nearest neighbor technique to detect anomalies in WSNs. This work relies on a previous work of us [100] described in Chapter 3. We model time series forecasting in WSNs and we term outliers the values that do not respect the model of data. We aim for energy efficient data collection and communication. In the following section, we discuss the contribution and motivation of our work.

5.3 Motivation and Contribution

Sensor nodes rely on each other in their sensing, data transmission, and decision-making prcesses. They collaborate in monitoring and reporting observed events. However, data reported need to be of good quality. It should be accurate and sent on time in order to allow for appropriate actions in a timely manner to serve an application requirements. The accuracy of data does not only increase the reliability of the network. It may have its influence on the energy resources of nodes especially for resource constraint ones. Actually, the accumulation of undetected abnormal values during data collection drain nodes performance in their behaviors as well as their lifetime. We are concerned on observing accuracy among data in order to reveal abnormal behaviors. Several problems affect the accuracy of sensor data: hardware defects, environmental variations, missing sensor values, etc. Data generated by sensors observing the same phenomena may be temporally and spatially correlated. Whereas, outlying readings are geographically independent [132]. They are also values that do not conform with the normal distribution of data in time. Malicious attacks try to adapt themselves to the reading and be masked.

In our work, we try to detect temporal abnormal behaviors. Our approach is based on exploiting the temporal relationships among data collected by sensors to ensure the reliability of this data and to detect outliers. This task is based on our previous work in [100], in which we use a *predictive time-series model* based on autoregressive models (AR). The future readings of a node are predicted from a number of previous history of sensed values. We consider the work in [100] efficient in reducing the communication traffic between sensors and their sink. It also maintains an acceptable prediction accuracy according to the data type and provides less resource consumption for sensors (refer to Chapter 3).

In order to adapt our local prediction model to variations in the data distribution, each sensor continuously maintains its local model. It monitors the variation and correlation of previous data to detect significant changes. The quality of the model is supervised by two evaluation constraints: an *error bound* above which the prediction model re-adjustment is needed; and a ranking conditions for *anomaly detection*. These evaluation constraints are determined based on previous history data behaviors and correlation.

Our model is similar to the work in [233] in that it uses auto-regressive prediction model built at each sensor node to reduce communications and detect anomalies. Authors rely on two thresholds (δ and ε) to distinguish between actions. If the absolute value of the prediction error falls in [0, δ], then the model is a good predictor of the data. If it falls in [δ, ε], the data is still within the user specified error bound but the model might need to be updated. Finally, if the error prediction exceeds ε, then the data is an outlier. Their approach differs from ours in that they use a user-defined prediction while we compute a dynamic error bound value [100] (see previous chapters). In addition, they consider an anomaly bound that is always greater then the re-learning bound which helps to incorrectly classify masked anomalies as normal. An anomaly may be "masked", in the sens that the difference between its value and the corresponding predicted one by the model, is less than the error bound. Figure 5.1, illustrates an example.

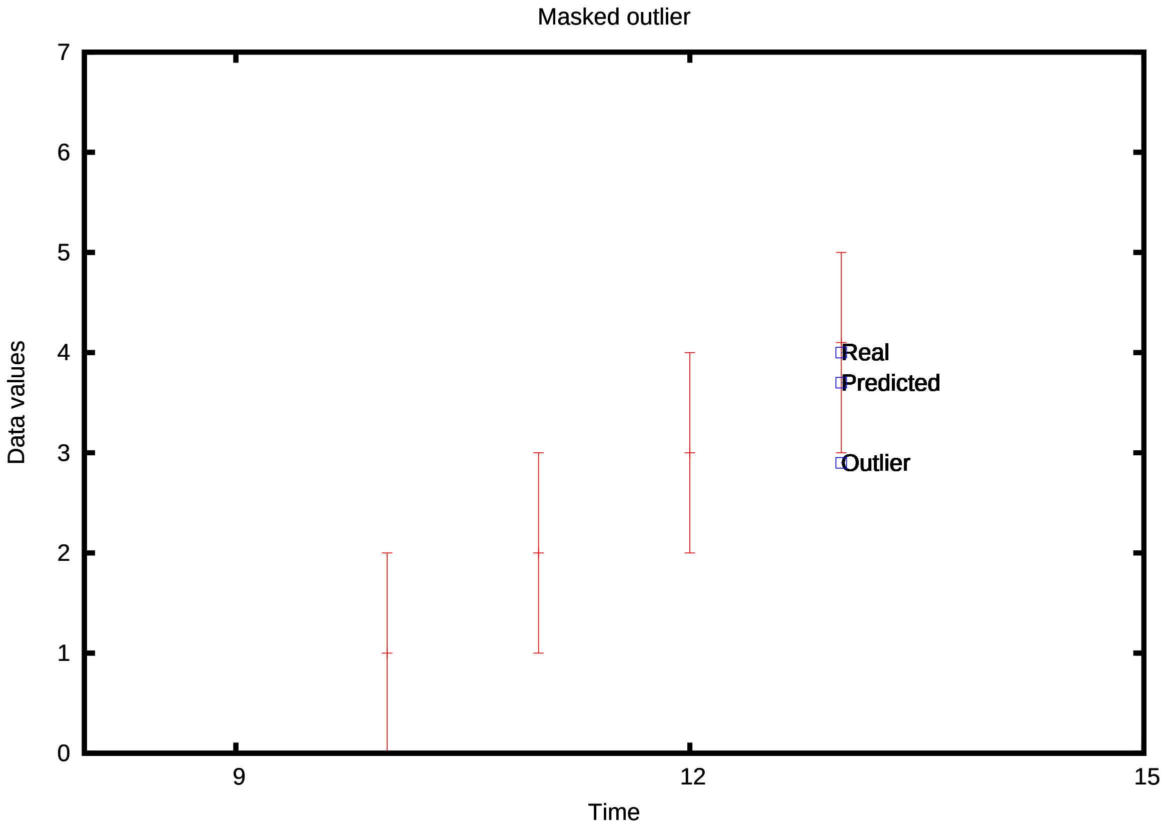


Figure 5.1: Example of masked anomalies.

Suppose having a data series $(N_i)_{i=0..3} = \{1, 2, 3, 4\}$. The error prediction bound $e = 1$ (based on the calculation proposed in [100]). One can produce an anomaly by signaling N_3 value as 2.9. If $X_3 = 3.7$, no event/anomaly will be detected since $|X_3 - N_3| \leq e$.

Thus 2.9 will be ranked as normal. Actually, repeating such behaviors forces the sensing operation and the prediction model to diverge from the real data distribution. Since re-learning the prediction model is based on previous values considered to be true.

The choice of a suitable error prediction bound and an anomaly bound is not a trivial task. In this work, we rely on our previous work to calculate a dynamic error bound [100] (EEE^* algorithm). As for anomalies detection, we do not rank each point on the basis of its distance to its q^{th} nearest neighbor as in distance-based approaches [187]. We do not also consider an anomaly bound greater of an error bound as in [233]. Since anomalous values, typically deviate from the normal data distribution, we observe different criterion: The weight of a point with respect to its previous data points (using kernel density estimation), the current spreading of each pair of observations as well as the chosen prediction error bound. In what follows, we define these criterion and present the notations in our algorithm.

5.4 Preliminaries

Along the process of sensing and predicting, each observed data point contributes−to some extent− in the estimation of future data points. This contribution depends on how much these points are apart. In order to understand the relation between data points, their dispersion and the influence of each one among the distribution, we associate to each observation N_t at time t, two attributes denoted by $dens(t)$ and $sp(t)$. In this section, we describe the symbols and notations used by our algorithm in this chapter.

- N_t: An observed value at time instant t.
- X_t: The prediction value of N_t.
- k: The order of prediction model.
- q: A rank for closest neighbor.
- The popular gaussian kernel used to measure similarity between X_t and X_p:

$$K(\frac{X_t - X_p}{h}) = \frac{1}{\sqrt{2\pi}} \exp \frac{-(X_t - X_p)^2}{2 * h^2} \tag{5.1}$$

- h: The width of the Gaussian kernel (we set $h = 1.74$[200]).
- The kernel density estimation of N_t among its k recent readings:

$$dens(t) = \frac{1}{k \times h} \sum_{p=t-1}^{t-k} K(\frac{X_t - X_p}{h}) \tag{5.2}$$

- The expected deviation of N_t from its previous data neighbors:

$$exp(t) = \sum_{j=t-1}^{t-k} dens(j) \times sp(j) \tag{5.3}$$

- The average distance between pair of points from N_t to its k previous data points considered as normal (the spreading function):

$$sp(t) = \sum_{p=t}^{t-k} \frac{|N_p - N_{p-1}|}{k} \tag{5.4}$$

We denote by $dens(t)$ (eq.((5.2))), the kernel density estimation, to evaluate the density of N_t within the distribution. In this chapter, we will adopt the popular gaussian kernel used in many applications domains [87, 198, 204].

We denote by $sp(t)$ (eq.((5.4))) the average distance between pair of points from N_t to its k previous data points considered as normal. We will use these functions in our proposed method for anomaly detection. In addition, we estimate for each observed value N_t its deviation $exp(t)$ from its previous data neighbors. The expected deviation (eq.((5.3))) is computed using the process of convolution (aggregation of coefficients) between its density and the spreading function. The purpose of doing this is to track the effect of past observations, their distribution and their density to the current observed value. In our algorithm, a value is considered anomalous if its deviation and density with respect to other points do not agree with the expected benchmarks of our algorithm. Our anomaly detection technique is described in the next section.

5.5 Our anomaly detection technique

Developing a model that facilitates the representation of sensor data and ensures a desired level of accuracy, presents many challenges. In fact, the boundary between normal and outlying behavior is often not precise. For anomaly detection algorithms in WSNs, the key challenge is to maintain an accurate anomaly detection rate, a lower false alarm rate as well as the resource consumption.

Our algorithm relies on our previous work on time series forecasting [100] and focuses on the temporal anomaly detection. We aim at observing the temporal correlation among data, to distinguish between: *mispredicted* values, "*weird*" and *normal* behaviors at time instant t. The study of spatial correlation for anomaly detection will be integrated in our future work. For instance, we assume having a suitable spatial aggregation techniques and network topology. Our technique is composed of two main units: A *predictive modeling unit* (described in Chapter 3) and an *anomaly detection unit* described in what follows.

5.5.1 Detection criterias

The temporal anomaly detection is the main task of this chapter. A sensor marks a reading as outlier depending on the extent to which the reading disagrees with the data distribution history. In other words, an observed value N_t at time t is ranked as "normal" if it has these characteristics:

(i) If its degree of similarity with the previous k data points do not tend to zero.

(ii) If the current spreading $sp(t)$ (eq.((5.4))) of N_t with the previous k data points is less than the expected deviation $exp(t)$ (eq.((5.3))). Thus $sp(t) \leq exp(t)$.

Note that if an observation breaks the above ranking conditions, it is considered "weird/abnormal". The sensor then, moves to discuss with its neighbors the real state of this observation. In fact, the spatial anomaly detection is out of the scope of this work. Therefore, we assume having a suitable protocol for spatial correlation that is able to replace this value with the real one.

Calibration for ranking conditions:

The idea of condition calibration is motivated by the relation between the error prediction bound th, the current spreading of points $sp(t)$ and the expected deviation $exp(t)$ of the observed value N_t. Note that:

$$sp(t) = \frac{k-1}{k} \times th + \frac{|N_t - N_{t-1}|}{k} \pm r_{rand} \tag{5.5}$$

Since the generated values of th, $sp(t)$ and $exp(t)$ belong to a range of scores (an interval), we replace the condition in (ii) by intervals comparison.

We denote by $I_{sp(t)}$ and $I_{exp(t)}$, the estimate intervals of $sp(t)$ and $exp(t)$ at time t respectively. The condition (ii) becomes:

$$sp(t) \leq exp(t) \parallel I_{sp(t)} \subseteq I_{exp(t)} \tag{5.6}$$

In the other hand, $MaxBound(I_{exp(t)}) - exp(t)$ is the expected data spreading range. So in order to control the spreading of a predicted value from the previous k values, we add the following condition:

$$|N_t - X_t| \leq MaxBound(I_{exp(t)}) - exp(t) \tag{5.7}$$

5.5.2 Summary of our technique

The major steps of our algorithm are as follows:

1. Monitoring data distribution for anomaly detection: Normal versus intrusion classification is made by the following ranking conditions: (i), eq. (5.6) and eq. (5.7).

   a. If an observed value N_t at time t satisfies the ranking conditions, it is then classified as "normal". Then, the sensor moves to the next step.

   b. Otherwise, the sensor marks N_t as "weird/abnormal" and turns to settle the issue with its neighbors nodes to get the normal value.

2. Identification of model misprediction: Once an observed value N_t had passed the above ranking conditions, it will contribute to observe the prediction model behavior. Hence, EEE^+ algorithm is fired:

   a. If $e > th$, the model needs to be adjusted and a communication between the sensor and the sink occurs. Hence, previous prediction errors are sent to the sink based on [100]).

   b. If $e \leq th$, no need for communication.

5.6 Evaluation methodology and Setup

The key challenge of an anomaly detection algorithm is to maintain high detection rate while keeping low false alarm rate. This requires the construction of an accurate and representative normal profile, a task that can be very difficult for sensor network applications. In this section, we evaluate the efficiency of our algorithm on real data series in different fields: The chemical process temperature readings taken every minute, the monthly measurements of carbon dioxide above Mauna Loa[1], a garden temperature data [124] and cardiac frequency measurements[2].

5.6.1 Generating anomalous data

To slip abnormal behaviors in the data, we relied on the confidence interval to make the perturbation. Given σ and n, the population standard deviation and size, the confidence intervals $[\bar{\mu} \pm \frac{\sigma}{\sqrt{n}}]$, $[\bar{\mu} \pm \frac{1.96\sigma}{\sqrt{n}}]$ and $[\bar{\mu} \pm \frac{3\sigma}{\sqrt{n}}]$ are of confidence level 68%, 95% and 99% respectively. We evaluate our technique by choosing random anomalous data in $[\bar{\mu} \pm \frac{\sigma}{2\sqrt{n}}]$ to have lowest confidence level.

First, we slipped 50 anomalous values among the real observations on a positions of time in [4,226] with a random perturbation rate. Then we increased the outliers quantities. As sample, Figure 5.2 represents normal versus intrusive data for the chemical process temperature, as well as the random perturbation rate generating 50 intrusions.

5.6.2 Evaluation metrics

The effectiveness of anomaly-detection techniques is generally evaluated by their ability to distinguish between normal and abnormal behaviors. An approach's ability to correctly classify behavior is essentially interpreted in terms of four possibilities:

- True-positive (TP) events that are attacks instances detected as abnormal.
- False-positive (FP) are events incorrectly classified as abnormal.
- True-negative (TN) are events correctly identified normal behavior.
- False-negative (FN) are abnormal behavior incorrectly classified as normal.

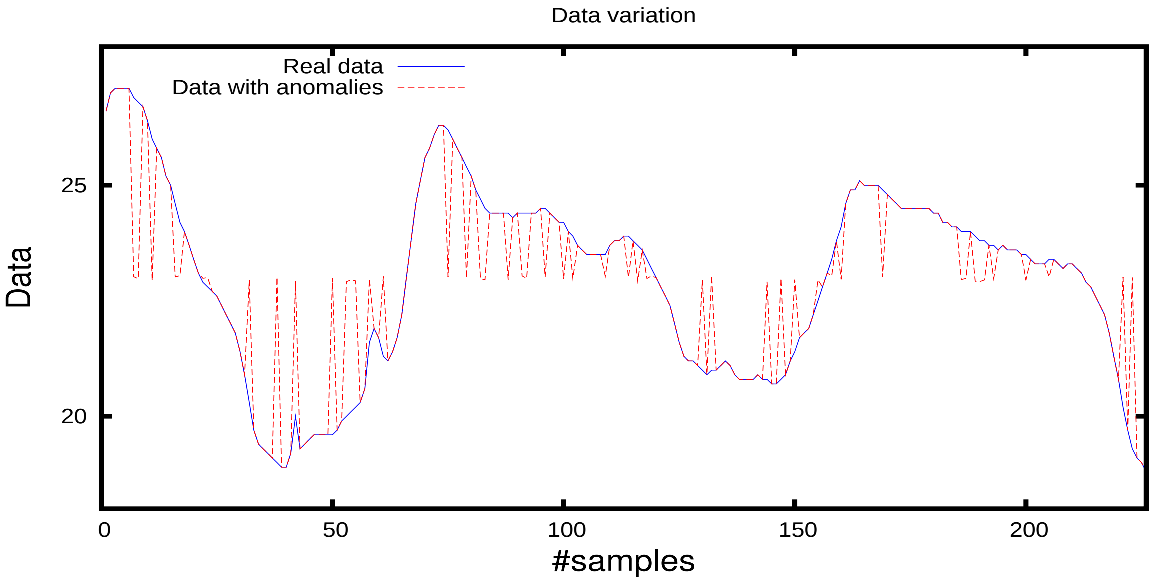
To evaluate these possibilities, two metrics were used. These are the most commonly used metrics in research work (see [226]):

- The detection rate (DR): a ratio between the number of correctly detected attacks and the total number of attacks.
- The false positive rate (FPR): computed as the ratio between the number of normal connections that are incorrectly misclassified as attacks and the total number of normal connections.

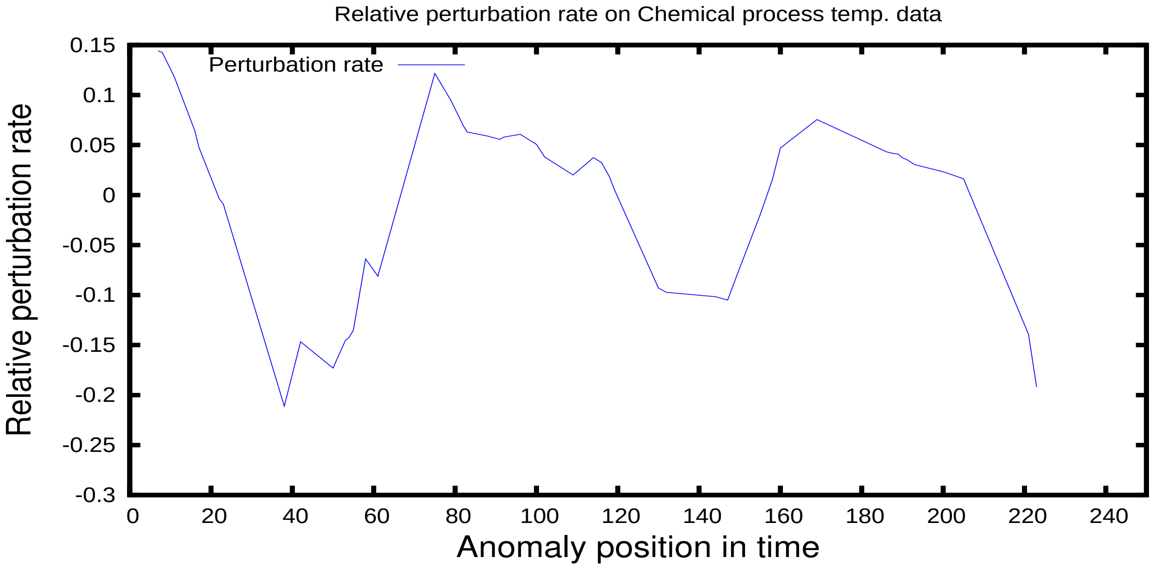
  Often these metrics are displayed as a receiver operating characteristic (ROC) curve

[1] http://www.robjhyndman.com/TSDL/

[2] http://www.inrialpes.fr/Xtremlog/



(a) *Chemical process T°*



(b) *Random Perturbation rate on Chemical process T°data*

Figure 5.2: Anomalous v/s normal data and the random perturbation rate generating 50 outliers for the chemical process temperature.

to emphasize the tradeoff between them. We will use these metrics to evaluate our technique.

5.6.3 Results and performance

- *Case study of 50 anomalous values*: The performance of our algorithm concerning the detection rate (DR) is presented in Table 5.1 for different data types. It is 92% for CO2 measurements, about 78% for garden temperature, 90% for chemical process temperature, and up to 80% for cardiac frequency measurements (which is high detection rate). As for the false-negative (FN) events. Our experimentation results show that (FN) values are the measurements obtained by a very low perturbation rate on the real data values. This perturbation rate made them undetected by our algorithm. Figures 5.3 and 5.4 show the relative perturbation rate of the undetected anomalous values for different data types. It indicates a perturbation rate less then 5%. We then consider our algorithm efficient with high anomalies detection rate.

	CO2	Garden T°	Cradiac Freq.	Chemical T°
TPR (DR)	92%	78%	80%	90%
FPR	11%	5%	17%	8.5%

Table 5.1: TP rate and FP rate produced by our technique for each data type.

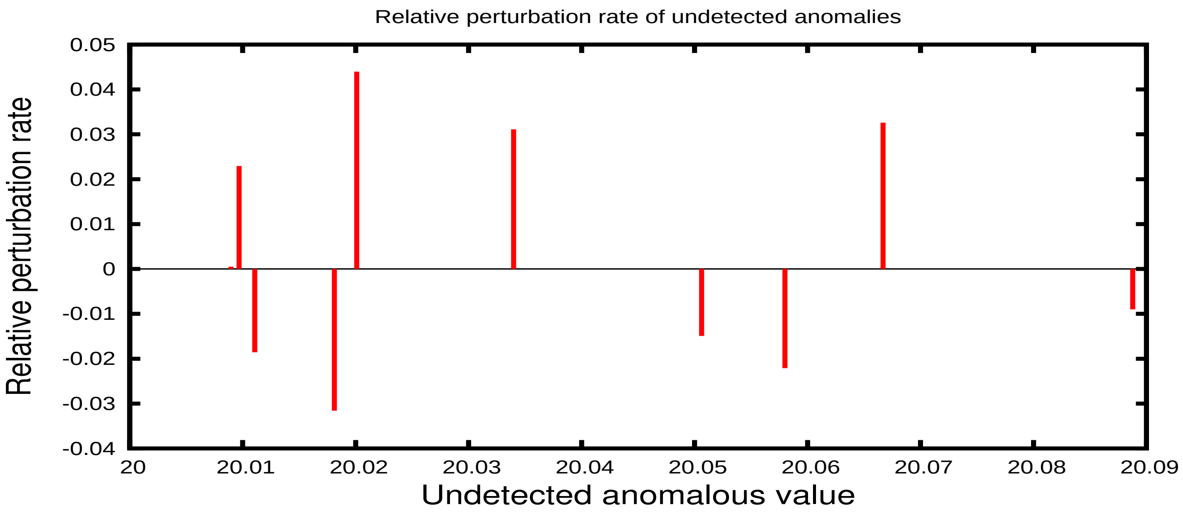
  The misclassification of data (false alarms) is one of the primary drawbacks of anomaly detection techniques. Therefore, we observe the false alarm rate in our experimentation. False alarms occur when normal behaviors are marked as anomalous. This can be related to the natural changes in the external observed phenomena (the real data behavior). False alarm rate may produce heavy effect on the energy consumption for WSNs. Table 5.1 shows the false alarm rate (FPR) of our technique. We clearly observe that the false alarm rate for garden and chemical process temperatures is low and is about 5% and 8% respectively. Our previous work in [100] (Chapter 3) shows that our prediction model is efficient in terms of energy consumption. It produces about 70% reduction in terms of communication traffic for some data types. After the extension of that work to an anomaly detection technique in this chapter, we noticed that the efficiency in terms of energy consumption did not decrease significantly for slow variation times series measurements (such as, garden and chemical process temperature measurements). This is due to a low false alarm rate detected about 8%.

  On the other hand, we noticed that our algorithm is not suitable for high variation time series measurements (such as cardiac frequency and CO2 measurements). The produced false alarm rate increases the communication cost and of course harvest energy.

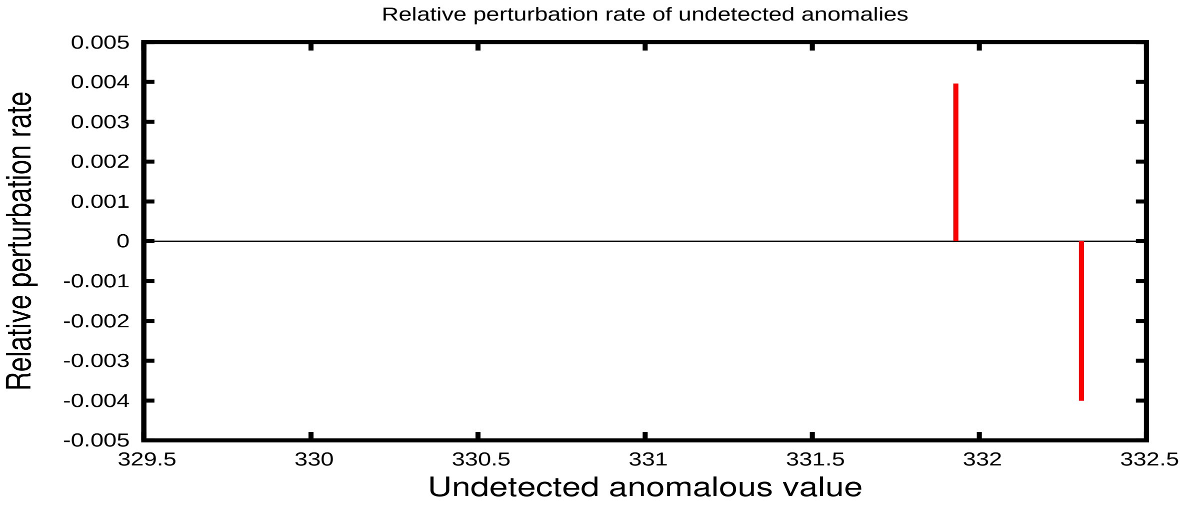
- *Detection accuracy*: To study the efficiency of our technique in terms of accuracy, we compared our results to TRAMO/SEATS[3] program results.

  TRAMO ("Time Series Regression with ARIMA noise, Missing values, and Outliers") is a program for estimation and forecasting of regression models as well as

[3] http://www.bde.es



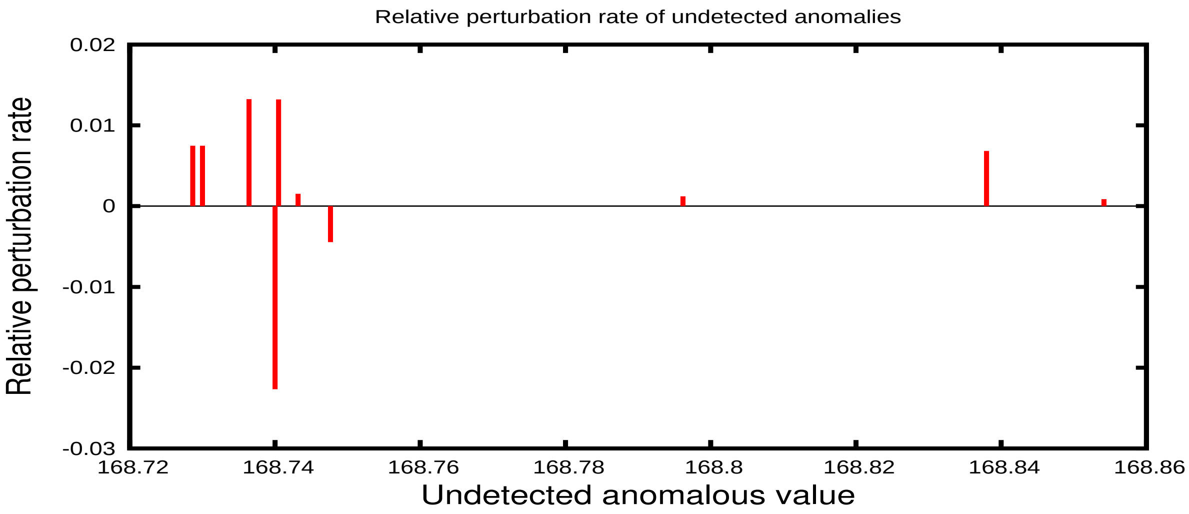
(a) *Garden temperature*



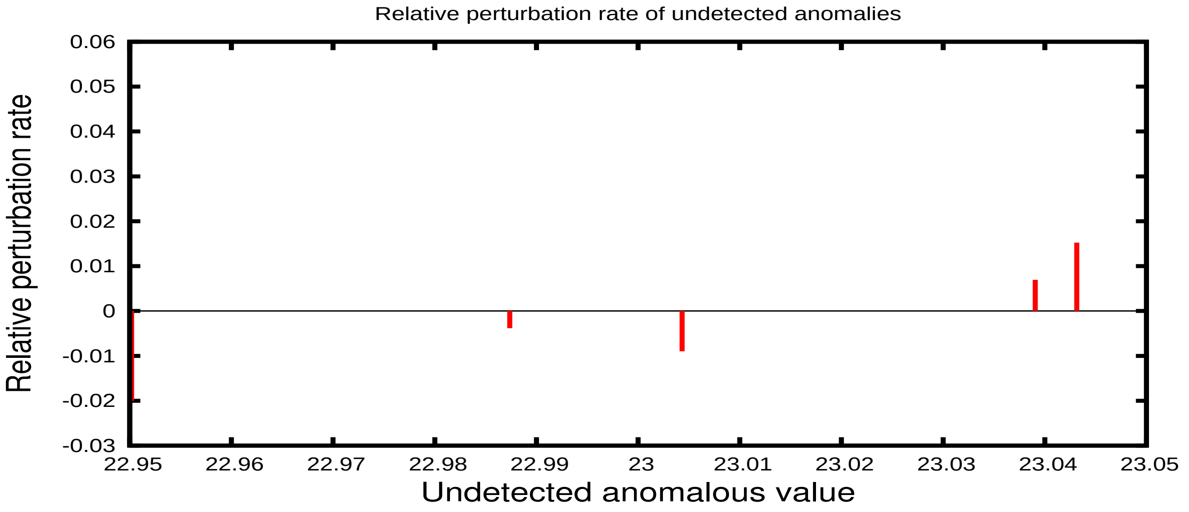
(b) *CO2*

Figure 5.3: The perturbation rate of undetected anomalous values for some data type.

identifying outliers. For reason of fairness, we fixed the outlier detection threshold to its default in TRAMO (which is 3.5 [236]) for each data type. We also tested our algorithm with a static error threshold, we chose it as the data standard deviation for each data type. Figures 5.5 and 5.6 show the results obtained by each algorithm. The ROC curves were smoothed after slipping different outliers quantities. Since the accuracy of detection is indicated by the rise or "bowness" of the ROC



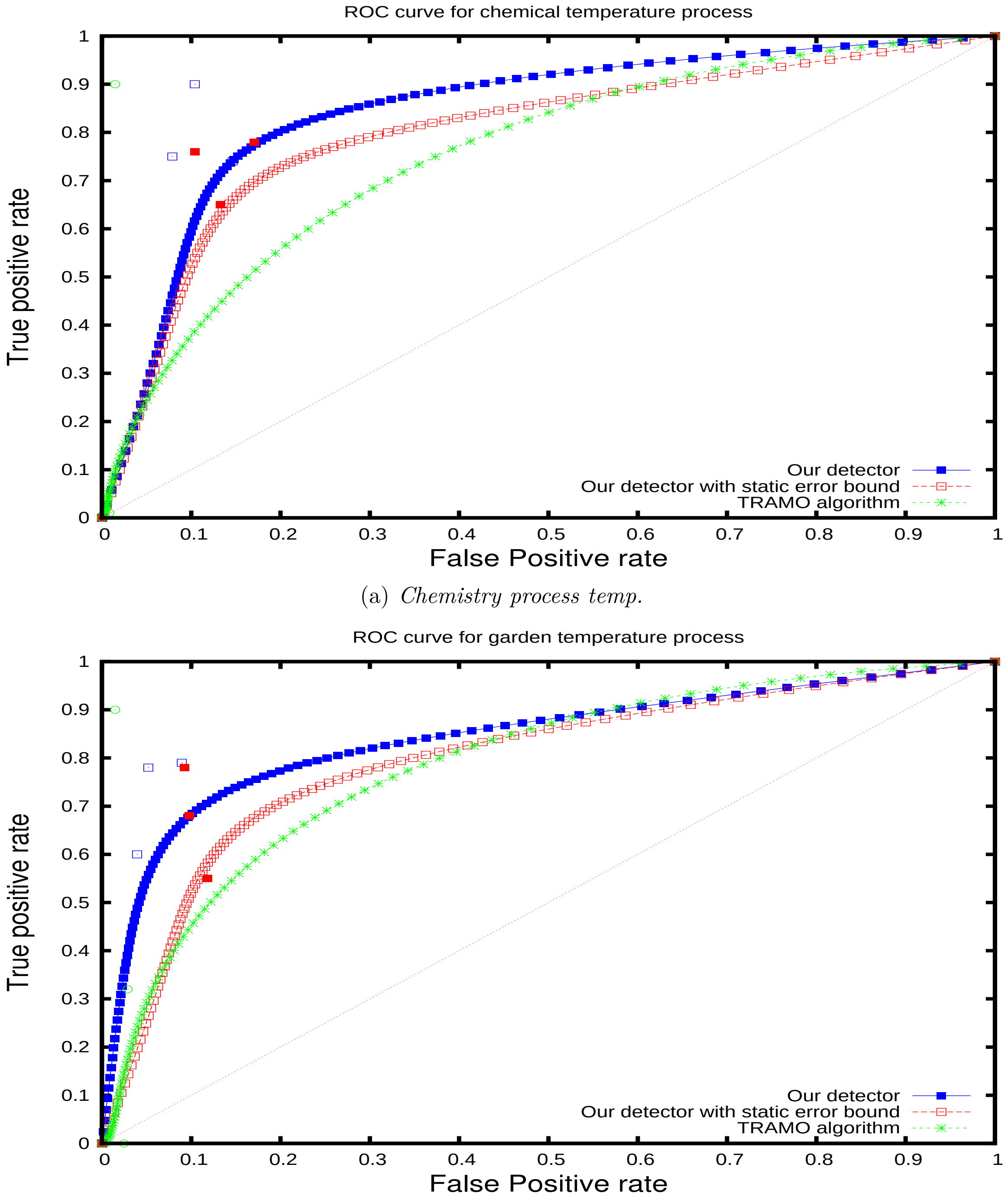
(a) *Cardiac frequency*



(b) *Chemistry process temp.*

Figure 5.4: The perturbation rate of undetected anomalous values for additional data type.

curve, the perfect ROC curve passes through the point (0,1). Among the different curves shapes, we notice that our algorithm produces higher detecting accuracy.



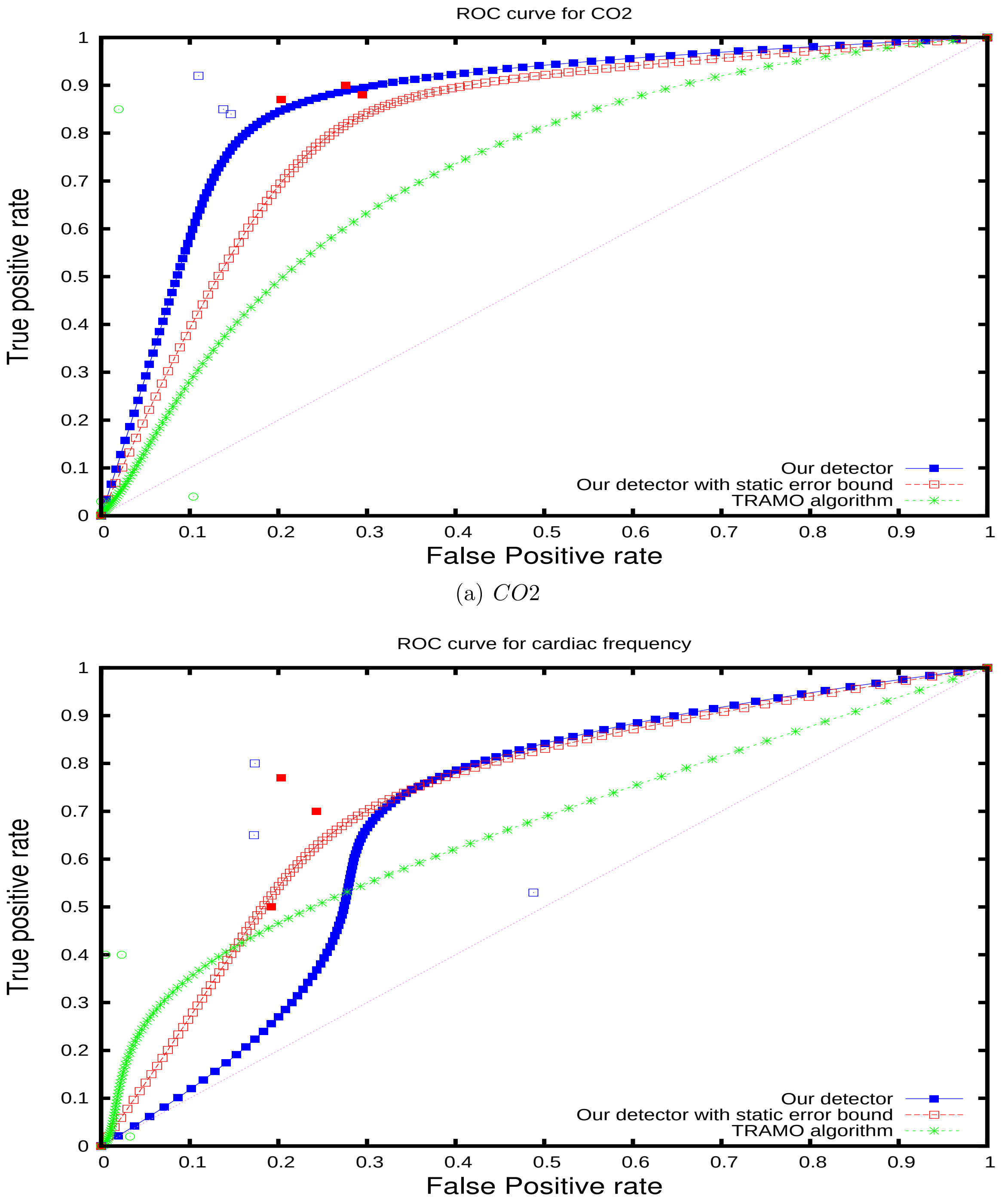
(a) *Chemistry process temp.*

(b) *Garden temperature*

Figure 5.5: ROC curves for Chemistry process temp. and Garden temperature data.

5.7 Conclusion

Detecting and reporting abnormal behaviors in sensor nodes is one of the main challenges in WSNs. Whereas sensors may have limited resource capabilities, data coming from different streams need to be examined dynamically for any occurrence of abnormality. It is important to process as much of the data as possible in a decentralized fashion (locally



(a) *CO2*

(b) *Cardiac frequency*

Figure 5.6: ROC curves for CO2 and Cardiac frequency data.

on nodes), so as to avoid unnecessary communication costs. However, the construction of an accurate and representative normal profile of data that is able to distinguish weird and normal behavior is not trivial. Our work first approximates the data distribution dynamically from a set previous values. We use a prediction model to estimate an event behaviors over time (temperature, humidity, etc.). Second, we observe the evaluation of data over time to differentiate between mispredicted values, abnormal and normal ones. For this purpose, we formulate a set of criteria for detection or ranking conditions from a history of normal behaviors. In our technique we concentrate our efforts on the accuracy of both the data prediction and the abnormality detection. Our anomaly detection technique relies on the prediction model proposed in Chapter 3 and tries to detect temporal anomalous values.

The experimentation results show that our algorithm produces high detection rate of abnormal behaviors. It can accurately detect between 78% up to 90% of slipped anomalies. It also produces low false alarm rate (∼8%) for slow variation data types. Although it produces a good accuracy detection, the weakness of our algorithm is that it is not suitable for high variation time series data. Since it produces about 17% of false alarm rate which increases the energy consumption. In this chapter, we studied simple events that require only participation of a single sensor (local temporal detection). It is an initial step for spatio-temporal anomaly detection that will be a part of our future works.

If we knew what it was we were doing, it would not be called research, would it?
$\sim\sim\sim\sim$*Albert Einstein*

You should never be ashamed to admit you have been wrong. It only proves you are wiser today than yesterday.
$\sim\sim\sim\sim$*Jonathan Swift*

Chapter 6

Conclusions and Future works

Summary

In this chapter, we summarize the conclusions and insights we have discussed throughout the thesis and present some future research directions.

Contents

6.1 Outline of results

A fundamental challenge in the design of Wireless Sensor Networks (WSNs) is to maximize their lifetime. In this thesis, we observe the issue of saving energy by reducing the transmission rate and overhead through data fusion. Although data fusion conserves energy, it can lose much of the original structure in the data during compression, aggregation, etc. Such looseness affects the quality of information constructed at the base station. Therefore, we observe the performance of the network in terms of quality of collected data, as well as the energy consumption. We propose a suit of techniques including time series forecasting, similarity measurements and outliers detection techniques. The core idea of these techniques is to decrease the transmission cost (overhead/rate) of nodes towards the base station by:

- Achieving an accurate data prediction at the node/sink level. We use an AR-based model for prediction, to avoid high communication rate and to keep data consistency.
- Detecting and reducing redundancy among the transmitted data by observing similarity of measurements. Data aggregated at the intermediate nodes are checked for spatial/temporal correlation, before being transmitted toward the destination. Kernel-based methods were used for this purpose.
- Detecting anomalous values with high accuracy to maintain data reliability. A set of ranking rules were proposed to distinguish between normal and abnormal behaviors.

In [96, 99, 100], we propose an estimation based-AR (AutoRegressive) model to reduce energy spent for communication. Nodes prefer to predict data accuratly, to avoid periodic or unnecessary energy consumption. Three algorithms for Energy-Efficient Estimation were proposed (EEE, EEE^+ and EEE^* algorithms). We use static and dynamic bound to evaluate their estimation quality. Based on the experiments results, EEE^* shows a possibility for an accurate and energy-efficient estimation. This algorithm uses a small prediction window, as well as a dynamic learning for the prediction bound. It was compared in its efficiency to other models, such as AR/ARMA/A-ARMA [29, 40, 124]. Results show that, depending on data type, the transmission overhead and rate can be reduced (about 70% reduction in transmission rate [100]). A considerable prediction accuracy can be also obtained, since the relative errors for prediction were between 2% and 8%.

Our thesis proposes additional algorithm to reduce the transmission overhead [97, 100]. It observes similarity aspects among collected data through the spatio-temporal correlations. In this algorithm, we perform a cluster-based data aggregation and focus on reducing the transmission overhead during intra-/inter clusters aggregation. The algorithm uses the idea of "similarity regions". Each region is composed of different clusters expecting to have spatial data correlation. It takes advantage of the similarity measurements to detect data correlations and to reduce the transmission overhead. The experiments show a transmission overhead reduction of about 20%, in most used data types. As for the quality of data constructed at the base station, accurate results were revealed. In fact, the relative errors between the source and sink estimations were in $[-7 \times 10^{-7}, 10^{-6}]$.

We also tackle the temporal anomaly detection in our work. The algorithm proposed aims at detecting anomalies based on some ranking conditions. These conditions classify

a data point as abnormal based on its consistency with other neighboring points in the data distribution [98]. For the evaluation, we generate a random perturbation rate on the data and slipped a number of anomalous values. The performance of outliers detection was reflected by its high detection rate with low false alarms. The proposed algorithm can accurately detect between 78% up to 90% of slipped anomalies. It also produces low false alarm rate (∼8%) for slow variation data types.
Our thesis was a small contribution in the wide sensor networks research area and specifically for the energy consumption related studies. In what follows, we present some enhancement and future related works.

6.2 Future works

6.2.1 Short and medium terms

In the short term, we will continue to embed other analysis to examine our algorithms performance and reveal their limitations. Proposing a coding and representation schemes for the data transmitted in our algorithms, will help to evaluate the transmission cost as well as the energy consumption.

Later, our work can be enhanced by observing additional aspects for communication cost reduction. One of the main challenges of data aggregation and compression is to explore not only the correlation of data in time and space. In fact, exploring the similar conditions that nodes go through can also be useful. Sensors in similar environmental conditions, that are not necessarily spatially correlated, can report correlated data. For example, sensors nearby opened windows report high readings due to the external light. These readings should be similar to those sensors nearby light sources inside the building. Hence, correlation exists due to the similarity of environmental factors. Spatial correlation can be seen as one specific case of this because nearby nodes can have similar conditions. Investigating the data similarity by environmental factors, while building an efficient data aggregation techniques, will be among our medium term works. Such similarity aspects can improve the definition of the similarity regions and their contribution in the information fusion and communication cost reduction. Sensors can be classified in regions based on the rules for their similarity aspects. Such relations may be useful in tracking events and classifying objects behaviors. For example, monitoring animals behavior in labs, or human and plant reactions facing common conditions. We will investigate how to model data related by common conditions and how to learn (*i.e.* predict) data from similar regions. Another issue to consider is the possibility to automatically group sensors into regions, once the similarity conditions change. For example, sensors nearby the light sources will no longer belong to the same similarity regions as sensors nearby opened window, at night-time. The classification of data, however, faces different challenges. One of these challenges is to differentiate between normal and abnormal behaviors, that have changed the common conditions between sensors. The spatial correlation can be observed along with the similar conditions to produce accurate classification.

6.2.2 Long term

EEE^{*} and anomaly detection algorithms are suitable for low variations time series (as mentioned in the chapters before). However, this does not mean they cannot be enhanced for other kind of data distribution. In the contrary, the existence of a possible enhancement has its signs in the experimentations results. For example, 80% of slipped anomalous values in the cardiac frequency measurement can be detected while 17% of false alarm is produced. As for the prediction accuracy, the performance of EEE^{*} algorithm, was not bad for high variation time series measurement. 99% of relative errors for the cardiac frequency and the radiosity measurements were less then 5% and 8% respectively.

We foresee a possible enhancement with the extension of our work to integrate type-related data studies. For example: heart rate is related to blood pressure, the readings of humidity and barometric pressure sensors are related to the readings of the temperature sensors, and so on. Capturing type-related data (*i.e.* attributes) helps to improve information accuracy. However, this task is not trivial. The extraction of accurate and significant information, typically, requires to fuse and transform measurements from different sensors (*i.e.* heterogeneous sensors). Besides using data fusion, we will integrate knowledge extraction techniques from the fused data (inference, classification, etc.), to improve the quality of information. Prediction can not only be performed on the sequence of observed readings of an event, but also using the temporal relation information from other observed events. In-network knowledge aggregation is a high-level fusion. Exploring knowledge-based data aggregation will be amongst our future works.

A systematic study for efficiently extracting information/knowledge during in-network data aggregation is one of the promising strategies. It gains more importance, especially in the vision of ambient intelligence of the Internet of Things (IoT). In a smart home, for example, time series knowledge representation, can predict the temporal order of actions and objects behaviors. If a habitant enters a room, the light will turn on and if it is summer and it is hot inside the room, the AC will turn as well. An alarm can ring early after an analysis of the traffic and weather state. More complex information extraction can be made depending on the level of required knowledge. However, high level of information may require sophisticated analysis. Such analysis may drain energy for battery-operated sensor networks. The efficiency of information extraction in WSNs is related to the desired level of information and the network lifetime. One of the challenges is to observe the impact of knowledge-based data aggregation on the energy consumption. The selection of an aggregation point or the routing decisions may be influenced by the wisdom degree of a specified node. We mean by “wisdom”, the contribution degree of a node in the global knowledge extraction at the sink. The choice of the routing path, for example, may not only refer to the communication overhead, coverage and energy level of nodes. It will be interesting to observe an additional factor, that is the level of wisdom owned by nodes. A node may observe the event-relation with its neighboring nodes to perform energy-efficient in-network knowledge aggregation. One of the challenging issues is to select the routing path that aggregates most useful information with less energy consumption during knowledge aggregation.

Knowledge discovery from heterogeneous sensors occurs in many applications such as health care, smart home, sports, etc. In Body Sensor Networks (BSNs), the knowledge extraction poses several analytic challenges driven by the heterogeneous multi-modal nature. For example, BSNs in sports have been used for monitoring of player performance during training and for improving player techniques. For instance, TennisSense [133], a multi-modal sensing platform, uses a Wireless Inertial Monitoring Unit on the racquet arm of the player, combined with nine external networked digital video cameras, and a set of Ubisense 3D-tracking sensors around a tennis court, to review the performance of each player. Similar applications have also been developed for training snowboarders and golfers. There are several systems, communications, and sensor design challenges that need to be overcome. Importantly, body sensor data collected from different sources need to be supported by efficient techniques (classification, mining, analysis, retrieval and visualization techniques) to provide an appropriate knowledge extraction.

Bibliography

[1] *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, MASS 2009, 12-15 October 2009, Macau (S.A.R.), China*. IEEE, 2009.

[2] *2010 IEEE Wireless Communications and Networking Conference, WCNC 2010, Proceedings, Sydney, Australia, 18-21 April 2010*. IEEE, 2010.

[3] Wu A. and Abouzeid A.A. Error robust image transport in wireless sensor networks. *In 5th Workshop on Applications and Services in Wireless Networks (ASWN 2005)*, 2005.

[4] Jafar Adibi, Wei-Min Shen, and Eaman Noorbakhsh. Self-similarity for data mining and predictive modeling - a case study for network data. In *PAKDD*, pages 210–217, 2002.

[5] I. F. Akyildiz and I. H. Kasimoglu. Wireless sensor and actor networks: research challenges. 2(4), 2004.

[6] I. F. Akyildiz, T. Melodia, and K.R. Chowdhury. A survey on wireless multimedia sensor networks. Computer Networks.

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

[8] Yaqoob J. Y. Al-raisi and Nazar E. M. Adam. Monitoring wireless sensor network performance by tracking node operational deviation. 2010.

[9] Muhammad Mahtab Alam, Olivier Berder, Daniel Menard, Thomas Anger, and Olivier Sentieys. A hybrid model for accurate energy analysis of wsn nodes. *EURASIP Journal on Embedded Systems*, 2010.

[10] C. Alippi, G. Anastasi, C. Galperti, F. Mancini, and M. Roveri. Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications. Proceedings of IEEE International Workshop on Mobile Ad-hoc and Sensor Systems for Global and Homeland Security (MASS-GHS 2007).

[11] C Alippi, R Camplani, and C Galperti. Lossless compression techniques in wireless sensor networks: Monitoring microacoustic emissions. In *Int'l Workshop on Robotic and Sensors Envs*, pages 1–5, October 2007.

[12] J. Scott Armstrong and Fred Collopy. Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*, 8:69–80, 1992.

[13] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Awmugam, M. Nesferenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. volume 46, pages 605Ű–634, New York, USA, December 2004. Computer Networks Journal, Elsevier North-Holland, Inc.

[14] Azvan Cristescu Baltasar, Rãzvan Cristescu, Baltasar Beferull-lozano, and Martin Vetterli. On network correlated data gathering. In *IEEE InfoCom*, pages 2571–2582, 2004.

[15] Vic Barnett and Toby Lewis. Outliers in statistical data. *New York: John Wiley Sons*, 1994.

[16] Can Basaran and Kyoung-Don Kang. *Quality of Service in Wireless Sensor Networks.* Book Series Computer Communications and Networks, 2009.

[17] Stephanie Bell. Measurement good practice guide. a beginner's guide to uncertainty of measurement. 2(11), 1999.

[18] Luís M. A. Bettencourt, Aric A. Hagberg, and Levi B. Larkey. Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks. In *DCOSS*, pages 223–239, 2007.

[19] Vijay Bhuse and Ajay Gupta. Anomaly intrusion detection in wireless sensor networks. *Journal of High Speed Networks*, 15:33–51, 2006.

[20] C. Bisdikian. On sensor sampling and quality of information: A starting point. *Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 279–284, 2007.

[21] C. Bisdikian, J. Branch, K.K. Leung, and R.I. Young. A letter soup for the quality of information in sensor networks. *IEEE International Conference on Pervasive Computing and Communications (PerCom'09)*, pages 1–6, 2009.

[22] V. Blanz, B. Schölkopf, H. Bultho, C. Burges, V. Vapnik, and T. Vetter. Comparison of view-based object recognition algorithms using realistic 3d models. In *Artificial Neural Networks ICANN'96*, volume 1112, pages 251–256. Springer, 1996.

[23] R. Blender, K. Fraedrich, and F. Lunkeit. Identification of cyclone-track regimes in the north atlantic. 123:727–741, 1997.

[24] Aggelos Bletsas. Evaluation of kalman filtering for network time keeping. *Proceedings of PerCom IEEE Internationa Conference on Prevasive Computing and Communications*, pages 289–296, March 2003.

[25] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[26] Boris R. Bracio, W. Horn, and D. P. F. Möller. Sensor fusion in biomedical systems. *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 3:1387–1390, 1997.

[27] Joel Branch, Boleslaw Szymanski, Chris Giannella, Ran Wolff, and Hillol Kargupta. In-network outlier detection in wireless sensor networks. In *ICDCS*, pages 51–58, 2006.

[28] Angelo Brayner, Aretusa Lopes, Diorgens Meira, Ricardo Vasconcelos, and Ronaldo Menezes. Toward adaptive query processing in wireless sensor networks. *Signal Processing Journal, Elsevier*, 87:2911–2933, 2007.

[29] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting.* Springer, March 2002.

[30] C. Brown, H. Durrant-Whyte, J. Leonard, B. Rao, and B. Steer. Distributed data fusion using kalman filtering: A robotics application. In *Data Fusion in Robotics and Machine Intelligence (Chapter 7), M. A. Abidi and R. C. Gonzalez, Eds. Academic Press, Inc.*, pages 267–309, San Diego, CA, 1992.

[31] Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S. Furey, Manuel Ares, Jr., and David Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. 97(1), 2000.

[32] Costas Busch, Rajgopal Kannan, and Athanasios V. Vasilakos. Quality of routing congestion games in wireless sensor networks. In *Proceedings of the 4th Annual International Conference on Wireless Internet*, WICON '08, pages 71:1–71:6, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[33] S. Capkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *Proceedings of the IEEE INFOCOM'05*, 2005.

[34] C. Cappiello and F. Schreiber.

[35] Haowen Chan and Adrian Perrig. Ace: An emergent algorithm for highly uniform cluster formation. In *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, pages 154–171, 2004.

[36] Tsz Ho Chan, Chi Keung Ki, and Hoilun Ngan. Real-time support for wireless sensor networks, 2005.

[37] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Outlier detection: A survey, 2007.

[38] Yao-Chung Chang, Z-S L, and Jiann-Liang Chen. Cluster based self-organization management protocols for wireless sensor networks. In *IEEE Trans. Consumer Electronics*, volume 52, pages 75–80, 2006.

[39] Chris Chatfield. A commentary on error measures. *International Journal of Forecasting*, 8(1):100–102, June 1992.

[40] Chris Chatfield. *The Analysis of Time Series.* CRC Press, July 2003.

[41] V. Chatzigiannakis and S. Papavassiliou. Diagnosing anomalies and identifying faulty nodes in sensor networks. In *IEEE Sensors Journal*, volume 7, pages 637–645, May 2007.

[42] Vasilis Chatzigiannakis, Symeon Papavassiliou, Mary Grammatikou, and Basil S. Maglaris. Hierarchical anomaly detection in distributed large-scale sensor networks. *Proceedings of ISCC*, 2006.

[43] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, 2001.

[44] Chiao-En Chen, Andreas M. Ali, and Hanbiao Wang. Design and testing of robust acoustic arrays for localization and enhancement of several bird sources. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, pages 268–275, 2006.

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

[46] Zhikui Chen, Song Yang, Liang Li, and Zhijiang Xie. A clustering approximation mechanism based on data spatial correlation in wireless sensor networks. 2010.

[47] Sen ching S. Cheung and Avideh Zakhor. Video similarity detection with video signature clustering. 2001.

[48] David Chu, Amol Deshpande, Joseph M. Hellerstein, and Wei Hong. Approximate data collection in sensor networks using probabilistic models. In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, 2006.

[49] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press, 2001.

[50] Matrix Computations. G. golub and c. van loan. 1989.

[51] Diane J. Cook and Sajal K. Das. Smart environments: Technology, protocols and applications. In *Wiley-Interscience*, 2005.

[52] National Research Council. Embedded, everywhere: A research agenda for networked systems of embedded computers. In *National Academy Press*, 2001.

[53] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[54] Wang CQ, Chen JM, and Sun YX. Sensor network localization using kernel spectral regression. *Wireless Communications and Mobile Computing*, pages 1045–1054, 2010.

[55] Razvan Cristescu, Baltasar Beferull-Lozano, and Martin Vetterli. Networked slepian-wolf: Theory and algorithms. In *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, pages 44–59, Berlin, Germany, January 2004.

[56] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines.* 2000.

[57] Paul Crook and Gillian Hayes. A robot implementation of a biologically inspired method for novelty detection. In *Proceedings of Towards Intelligent Mobile Robots Conference (TIMR)*, 2001.

[58] Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. 2, 1990.

[59] Daniel-Ioan Curiac, Constantin Volosencu, Dan Pescaru, Lucian Jurca, and Alexa Doboli. Redundancy and its applications in wireless sensor networks: A survey. *WSEAS Transactions on Computers*, 8(4), 2009.

[60] C.X.Fan and et al. *Communication principles.* Defense Industrials Press, 1984.

[61] Gislason D. Zigbee wireless networking. *Newnes*, August 2008.

[62] Ana Paula R. da Silva, Marcelo H. T. Martins, Bruno P. S. Rocha, Antonio A. F. Loureiro, Linnyer B. Ruiz, and Hao Chi Wong. Decentralized intrusion detection in wireless sensor networks. In *Proceedings of the 1st ACM international workshop on quality of service & security in Wireless and mobile networks*, pages 16–23. ACM Press, 2005.

[63] Xiaohua Dai, Feng Xia, Zhi Wang, and Youxian Sun. A survey of intelligent information processing in wireless sensor network. *MSN*, 2005.

[64] Belur V. Dasarathy. Information fusion - what, where, why, when, and how? editorial. *Information Fusion*, 2(2):75–76, 2001.

[65] Dennis DeCoste, Bernhard Schölkopf, and Nello Cristianini. Training invariant support vector machines, 2002.

[66] U.S. DEPARTMENT OF DEFENSE. *Data fusion lexicon.* Published by Data Fusion Subpanel of the Joint Directors of Laboratories. Tecnichal Panel for C3 (F.E. White, Code 4202, NOSC, San Diego, CA), 1991.

[67] Antonios Deligiannakis, Yannis Kotidis, and Nick Roussopoulos. Processing approximate aggregate queries in wireless sensor networks. *Information Systems Journal*, 31:770–792, December 2006.

[68] Amol Deshpande, Carlos Guestrin, Samuel Madden, Joseph M. Hellerstein, and Wei Hong. Model-driven data acquisition in sensor networks. In *VLDB*, pages 588–599, 2004.

[69] Peter Desnoyers, Deepak Ganesan, Huan Li, Ming Li, and Prashant Shenoy. Presto: A predictive storage architecture for sensor networks. *10th Workshop on Hot Topics in Operating Systems (HotOS X)*, June 2005.

[70] Maria J. Diamantopoulou, Elias Milios, Dimitrios Doganos, and Ioannis Bistinas. Artificial neural network modeling for reforestation design through the dominant trees bole-volume estimation. 22(4), 2009.

[71] D. Diamond. Energy consumption issues in chemo/biosensing using wsns. Energy and Materials: Critical Issues for Wireless Sensor Networks Workshop.

[72] Min Ding, Dechang Chen, Kai Xing, and Xiuzhen Cheng. Localized fault-tolerant event boundary detection in sensor networks. In *Proceedings of IEEE Conference of Computer and Communications (INFOCOM)*, pages 902–913, 2005.

[73] Min Ding, Xiuzhen Cheng, and Guoliang Xue. Aggregation tree construction in sensor networks. *IEEE Vehicular Technology Conference - VTC*, 4(4):2168–2172, October 2003.

[74] L. Doherty, B. A. Warneke, B. E. Boser, and K. Pister. Energy and performance considerations for smart dust. In *International Journal of Parallel Distributed systems and Networks*, volume 4, pages 121–133, 2001.

[75] Jiang Dong, Dafang Zhuang, Yaohuan Huang, and Jingying Fu. Advances in multi-sensor data fusion: Algorithms and applications, 2009.

[76] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, CIKM '98, pages 148–155, New York, NY, USA, 1998. ACM.

[77] Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. Biological sequence analysis: probabilistic models of proteins and nucleic acids. cambridge univ, 1998.

[78] Felemban E., C.-G. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic qos guarantee in reliability and timeliness domains in wireless sensor networks. *In Proceedings of IEEE INFOCOM*, 2005.

[79] Eiman Elnahrawy and Badri Nath. Context-aware sensors. In *EWSN*, pages 77–93. Springer-Verlag, 2004.

[80] H. Jair Escalante. A comparison of outlier detection algorithms for machine learning, 2005.

[81] E. Eskin. Anomaly detection over noisy data using learned probability distributions. *In: Proceedings of Machine Learning*, 2000.

[82] P. Levis et al. The emergence of networking abstractions and techniques in tinyos. In *In Proceedings of NSDI*, March 2004.

[83] V. Naik et al. Sprinkler: A reliable and energy efficient data dissemination service for wireless embedded devices. In *26th IEEE Real-Time Sys. Symp.*, December 2005.

[84] Lei Fang, Wenliang Du, and Peng Ning. A beacon-less location discovery scheme for wireless sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, pages 161–171, 2005.

[85] Elena Fasolo, Michele Rossi, Jòrg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14, April 2007.

[86] Elena Fasoloy, Michele Rossiy, Jörg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks: A survey. *IEEE Wireless Communications*.

[87] Maria florina Balcan, Avrim Blum, and Nathan Srebro. On a theory of learning with similarity functions. In *International Conference on Machine Learning*, pages 73–80, 2006.

[88] Randy Frank. Understanding smart sensors. In *Artech House Sensors Library*, 2000.

[89] Terrence S. Furey, Nello Cristianini, Nigel Duffy, David W. Bednarski, Michel Schummer, and David Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. 16(102000), 2000.

[90] Moncecchi G., Minel J.-L., and Wonsever D. A survey of kernel methods for relation extraction. 2010.

[91] Mohamed Medhat Gaber. *Data Stream Processing in Sensor Networks*. Springer, 2007.

[92] Raghu K. Ganti, Praveen Jayachandran, Haiyun Luo, and Tarek F. Abdelzaher. Datalink streaming in wireless sensor networks. In *Proceedings of the SenSys Conference*, pages 209–222. ACM, 2006.

[93] J. L. Gao. Analysis of energy consumption for ad hoc wireless sensor networks using a bit-meter-per-joule metric. In *IPN Progress Report*, pages 42–150, 2002.

[94] Tia Gao, Dan Greenspan, Matt Welsh, Radford R. Juang, and Alex Alm. Vital signs monitoring and patient tracking over a wireless network, 2005.

[95] Erol Gelenbe and Laurence Hey. Quality of information: an empirical approach. *In Proceedings of MASS*, 2008.

[96] Alia Ghaddar, Tahiry Razafindralambo, Isabelle Simplot-Ryl, David Simplot-Ryl, Samar Tawbi, and Abbas Hijazi. Investigating data similarity and estimation through spatio-temporal correlation to enhance energy efficiency in wsns. *Ad Hoc & Sensor Wireless Networks (AHSWN) journal (to appear)*, 2011.

[97] Alia Ghaddar, Tahiry Razafindralambo, Isabelle Simplot-Ryl, Samar Tawbi, and Abbas Hijazi. Algorithm for data similarity measurements to reduce data redundancy in wireless sensor networks. *International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 0:1–6, 2010.

[98] Alia Ghaddar, Tahiry Razafindralambo, Isabelle Simplot-Ryl, Samar Tawbi, and Abbas Hijazi. Algorithm for temporal anomaly detection in wsns. *IEEE Wireless Communication and Networking Conference (WCNC'11)*, pages 743–748, March 28-31 2011.

[99] Alia Ghaddar, Isabelle Simplot-Ryl, David Simplot-Ryl, Tahiry Razafindralambo, and Samar Tawbi. Algorithmes pour l'estimation des données dans les réseaux de capteurs. *11emes Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications (AlgoTel)*, pages 93–96, June 2009.

[100] Alia Ghaddar, Isabelle Simplot-Ryl, David Simplot-Ryl, Tahiry Razafindralambo, and Samar Tawbi. Towards energy-efficient algorithm-based estimation in wireless sensor networks. In *Proceedings of the 6th International Conference on Mobile Ad-hoc and Sensor Networks (MSN'10)*, pages 39–46, Hangzhou, China, October 2010. IEEE Computer Society.

[101] Mario Di Francesco Andrea Passarella Giuseppe Anastasi, Marco Conti. Energy conservation in wireless sensor networks: A survey. Ad Hoc Networks.

[102] Samir Goel and Tomasz Imielinski. Prediction-based monitoring in sensor networks: Taking lessons from mpeg. 2001.

[103] G.Padmavathi, P. Subashini, and M. Krishnaveni. A suitable segmentation methodology based on pixel similarities for landmine detection in ir images. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 1(5), November 2010.

[104] S Grime and M Stevens. Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5):849–863, 1994.

[105] Jingjing Gu and Songcan Chen. Manifold-based canonical correlation analysis for wireless sensor network localization. *Wireless Communications and Mobile Computing*, 2011.

[106] Carlos Guestrin, Peter Bodík, Romain Thibaux, Mark A. Paskin, and Samuel Madden. Distributed regression: an efficient framework for modeling sensor network data. In *IPSN*, pages 1–10, April 2004.

[107] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[108] John H. Halton. Sequential monte carlo techniques for the solution of linear systems. 9(2), 1994.

[109] M. Han, J. Xi, S. Xu, and F. L. Yin. Prediction of chaotic time series based on the recurrent predictor neural network. 52(12), December 2004.

[110] Douglas M. Hawkins. Identification of outliers. 1980.

[111] Ting He and Murtaza Zafer. Adaptive sampling for transient signal detection in the presence of missing samples. *1st international QoISN, (Part of IEEE MASS)*, 2008.

[112] Yuxin He and Paul G. Flikkema. System-level characterization of single-chip radios for wireless sensor network applications. In *Wireless and Microwave Technology Conference*, pages 1–5, April 2009.

[113] W.B. Heinzelman, A.L. Murphy, H.S. Carvalho, and M.A. Perillo. Middleware to support sensor network applications. *IEEE Network*, 18:6–14, 2004.

[114] Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1:660–670, 2002.

[115] Alfredo I. Hernandez, Guy Carrault, Fernando Mora, Laurent Thoraval, Gianfranco Passariello, and Jean-Marc Schleich. Multisensor fusion for atrial and ventricular activity detection in coronary care monitoring. *IEEE Transactions on Biomedical Engineering*, 46:1186–1190, October 1999.

[116] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David E. Culler, and Kristofer S. J. Pister. System architecture directions for networked sensors. In *Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems(ASPLOS IV)*, pages 93–104, Cambridge, MA, 2000.

[117] Y. Sankarasubramaniam E. Cayirci I.F. Akyildiz, W. Su. A survey on sensor networks. In *IEEE Communications Magazine*, volume 40, pages 104–112, 2002.

[118] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *MOBICOM*, pages 56–67. ACM, 2000.

[119] Tommi Jaakkola, Mark Diekhans, and David Haussler. A discriminative framework for detecting remote protein homologies, 1999.

[120] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press, 1998.

[121] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, Inc., 1988.

[122] Ankur Jain, Edward Y. Chang, and Yuan-Fang Wang. Adaptive stream resource management using kalman filters. pages 11–22, Paris, France, 2004. ACM.

[123] D. Janakiram, A. Mallikarjuna, V. Reddy, and P. Kumar. Outlier detection in wireless sensor networks using bayesian belief networks. *Proceedings of IEEE on Communication System Software and Middleware (Comsware)*, pages 1–6, 2006.

[124] Mischa Dohler Jialiang Lu, Fabrice Valois. Optimized data aggregation in wsns using adaptive arma. *SensorComm*, July 2010.

[125] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *In proceedings of the 10th European Conference on Machine Learning (ECML-98)*, 1998.

[126] Minwook C. Jun, H. Jeong, and Jay Kuo C.-C. Distributed spatio-temporal outlier detection in sensor networks. *Proceedings of SPIE*, 2006.

[127] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for smart dust. In *Proceedings of the 5th Ann. International Conference on Mobile Computing and Networking (Mobicom)*, pages 271–278, Seattle, WA, August 1999. ACM.

[128] Rajgopal Kannan, Sudipta Sarangit, S. S. Iyengar, and Lydia Ray. Sensor-centric sensor quality of routing in networks. In *In Proc. IEEE INFOCOM*, pages 692–701, 2003.

[129] N. Kim, S. Choi, and H. Cha. Automated sensor-specific power management for wireless sensor networks. In *Proc. IEEE Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, pages 305–314, 2008.

[130] L. A. KLEIN. Sensor and data fusion concepts and applications. *SPIE Optical Engineering Press*, 1993.

[131] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. pages 392–403, 1998.

[132] Bhaskar Krishnamachari and Sitharama Iyengar. Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Transactions on Computers*, 53(3):241– 250, 2004.

[133] Conroy L, Ciaran O Connaire, Coyle S, Graham Healy, Philip Kelly, Noel O'Connor, Caulfield B, Damien Connaghan, Alan F. Smeaton, and Nixon P. Tennissense: A multisensory approach to performance analysis in tennis. In *27th International Society of Biomechanics in Sports Conference 2009*, 2009.

[134] M-J. Lesot, M. Rifqi, and H. Benhadda. Similarity measures for binary and numerical data: a survey. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 1:63–84, December 2009.

[135] Xiaoxing Li. A survey on data aggregation in wireless sensor networks. *Project Report for CMPT 765*, Spring 2006.

[136] B. W. Lindgren. Statistical theory. 1993.

[137] Stephanie Lindsey, Cauligi Raghavendra, and Krishna M. Sivalingam. Data gathering algorithms in sensor networks using energy metrics. *IEEE Transactions on Parallel and Distributed Systems*, 13:924–935, 2002.

[138] Stephanie Lindsey, Cauligi S. Raghavendra, and Krishna M. Sivalingam. Data gathering algorithms in sensor networks using energy metrics. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):924–935, September 2002.

[139] Juan Liu, Ying Zhang, and Feng Zhao. Robust distributed node localization with error management. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*, pages 250–261, Florence, Italy, 2006. ACM.

[140] Konrad Lorincz, David Malan, Thaddeus R. F. Fulford-Jones, Alan Nawoj, Antony Clavel, Victor Shnayder, Geoff Mainland, Steve Moulton, and Matt Welsh. Sensor networks for emergency response: challenges and opportunities, 2004.

[141] Chenyang Lu, Brian M. Blum, Tarek F. Abdelzaher, John A. Stankovic, and Tian He. Rap: A real-time communication architecture for large-scale wireless sensor networks. In *IEEE Real Time Technology and Applications Symposium - (RTAS'02)*, pages 55–66, San Jose, CA, USA, 24-27 September 2002. IEEE, IEEE Computer Society.

[142] R. C. Luo and M. G. Kay. Data fusion and sensor integration: State-of-the-art 1990s. *Data Fusion in Robotics and Machine Intelligence*, pages Chapter 3, 7–135, 1992.

[143] Xuanwen Luo, Ming Dong, and Yinlun Huang. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on Computers*, 55:58–70, 2006.

[144] Xuanwen Luo, Ming Dong, and Yinlun Huang. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on Computers*, 55(1):58–70, 2006.

[145] U. luxburg. Statistical learning with similarity and dissimilarity functions, 2004.

[146] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[147] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. The design of an acquisitional query processor for sensor networks. *Proceedings of the International Conference on Management of Data*, pages 491–502, June 2003.

[148] N.S.R. Mardoqueu Souza Vieira. A reconŮgurable group management middleware service for wireless sensor networks. In *Proceeding of the 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing*, Grenoble, France, 2005.

[149] Markos Markou and Sameer Singh. Novelty detection: a review-part 1: statistical approaches. *Signal Processing*, 83, 2003.

[150] M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, A. Nadas, K. Molnar, S.Dora, and A. Ledeczi. Radio interferometric geolocation. In *Proceedings of the Third International Conference on Embedded Networked Sensor Systems (Sensys)*, San Diego, CA, 2005.

[151] Fernando Martincic and Loren Schwiebert. Distributed event detection in sensor networks. *Proceedings of the International Conference on Systems and Networks Communication*, pages 43–48, 2006.

[152] Cecilia Mascolo and Mirco Musolesi. Scar: Context-aware adaptive routing in delay tolerant mobile sensor networks. In *Proceeding of the 2006 International Conference on Communications and Mobile Computing (IWCMC'06)*, page 533Ű538, Vancouver, Canada, July 2006. ACM.

[153] F. R. McFadden, J. A. Hoffer, and M. B. Prescott. *Modern Database Management (Fifth ed.)*. Addison-Wesley, 1999.

[154] William Merrill, Katayoun Sohrabi, Lewis Girod, Jeremy Elson, and Fredric Newberg. Open standard development platforms for distributed sensor networks. In *In SPIE Unattended Ground Sensor Technologies and Applications IV*, pages 327–337, Orlando, FL, April 2002.

[155] Giacomo De Meulenaer, François Gosset, François xavier St, and Olivier Pereira. On the energy cost of communication and cryptography in wireless sensor networks. In *In Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB 2008)*, pages 580–585. IEEE Computer Society Press, 2008.

[156] Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, Alex Smola, and Klaus-Robert Müller. Invariant feature extraction and classification in kernel spaces. 2000.

[157] Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, and Klaus-Robert Müller. Fisher discriminant analysis with kernels. 1999.

[158] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rästsch. Kernel pca and de-noising in feature spaces. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 11*, pages 536–542. MIT Press, 1999.

[159] H. B. Mitchell. *Multi-sensor data fusion.*

[160] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the SensysŠ04*, San Diego, CA, 2004.

[161] F. A. Mora, G. Passariello, G. Carrault, and J. P. Le Pichon. Intelligent patient monitoring and management systems:a review. *In Engineering in Medicine and Biology Magazine.*

[162] S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear prediction of chaotic time series using a support vector machine. 1997.

[163] Eduardo F. Nakamura, Fabiola G. Nakamura, Carlos M. S. Figueiredo, and Antonio A. F. Loureiro. Using information fusion to assist data dissemination in wireless sensor networks. *Telecommunication Systems*, pages 237–254, November 2005.

[164] Eduardo Freire Nakamura, Antonio Alfredo Ferreira Loureiro, and Alejandro César Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.*, 39(3), 2007.

[165] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network.

[166] Hüseyin Òzgùr Tan and Ibrahim Kòrpeoglu. Power efficient data gathering and aggregation in wireless sensor networks. *SIGMOD Record*, 32(4):66–71, December 2003.

[167] Themistoklis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. Distributed deviation detection in sensor networks. *ACM Special Interest Group on Management of Data*, pages 77–82, 2003.

[168] Boaz Patt-Shamir. A note on efficient aggregate queries in sensor networks. *Theoretical Computer Science*, 370:254–264, 2007.

[169] Sundeep Pattem, Bhaskar Krishnamachari, and Ramesh Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. April 2004.

[170] S. Pattern, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. 2004.

[171] Neal Patwari, Alfred O. Hero, Matt Perkins, Neiyer S. Correal, and Robert J. O'Dea. Relative location estimation in wireless sensor networks. In *IEEE Trans. Sig. Proc.*, volume 51, pages 2137–2148, August 2003.

[172] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks.

[173] G. Pison, P. J. Rousseeuw, P. Filzmoser, and C. Croux. A robust version of principal factor analysis, 2000.

[174] Carlos Pomalaza-Ráez. *Wireless Sensor Networks.* University of Oulu, Finland, 2004.

[175] Gregory J. Pottie and William J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[176] A. Prati, R. Vezzani, L. Benini, E. Farella, and P. Zappi. An integrated multi-modal sensor network for video surveillance. Proceedings of the Third ACM international Workshop on Video Surveillance and Sensor Networks (VSSN'05).

[177] M. Pringle, T. Wilson, and R. Grol. Measuring "goodness" in individuals and healthcare systems. *British Medical Journal*, 325:704–707, 2002.

[178] H. Qi and F. Wang. Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks. *Proceedings of the13th International Conference on Wireless Communications (Wireless'2001)*, 2001.

[179] Liang Qilian and Wang Lingming. Event detection in sensor networks using fuzzy logic system. *IEEE Conference on Computational Intelligence for Homeland Security and Personal Safety*, I, March 2005.

[180] L. Qiong, W. Hejun, X. Wenwei, and H. Bingsheng. Benchmarking in-network sensor query processing, 2005.

[181] Clarke R., Waddington J., and J. N. Wallace. The application of kalman filtering to the load/pressure control of coal-fired boilers. *IEE Colloquium on Kalman Filters: Introduction, Applications and Future Developments*, 27:2/1–2/6, February 1989.

[182] Prasantht R., Cabrera J., Amin J., Mehra R., Purtell R., and Smith R. Quality of information measures for autonomous decision-making. *Proceedings of American Control Conference*, 2004.

[183] Jan M. Rabaey, M. Josie Ammer, Julio L. da Silva Jr., Danny Patel., and Shad Roundy. Picoradio supports ad hoc ultra low-power wireless networking. 33(7), July 2000.

[184] V. Raghunathan, S. Ganeriwal, and M. Srivastava. Emerging techniques for long lived wireless sensor networks. IEEE Communications Magazine.

[185] Ramesh Rajagopalan and Pramod K. Varshney. Data-aggregation techniques in sensor networks: A survey. *IEEE Communication Surveys and Tutorials*, 8:48–63, December 2006.

[186] Sutharshan Rajasegarar, Christopher Leckie, and Marimuthu Palaniswami. Distributed anomaly detection in wireless sensor networks. In *in Proceedings of Tenth IEEE International Conference on Communications Systems (IEEE ICCS 2006)*, 2006.

[187] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438, 2000.

[188] Dongmei Ren, Baoying Wang, and William Perrizo. Rdf: a density-based outlier detection method using vertical data representation. pages 503–506, 2004.

[189] K. Romer and F. Mattern. The design space of wireless sensor networks. In *IEEE Wireless Communications*, volume 11, pages 54–61. Technical Report. University of Maryland at College Park, 2004.

[190] P.J. Rousseeuw and A .M. Leroy. *Robust regression and outlier detection.* 1996.

[191] Virginia K. Saba and K. A. McCormick. *Essentials of Computers for Nurses. Informatics for the New Millennium (Third ed.).* New York: McGraw-Hill, 2001.

[192] P. Santi. Topology control in wireless ad hoc and sensor networks. In *John Wiley & Sons, Ltd*, 2005.

[193] Silvia Santini and Kay Romer. An adaptive strategy for quality-based data reduction in wireless sensor networks. *Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS2006)*, pages 29–36, 2006.

[194] Thorsten Schmitt, Robert Hanek, Michael Beetz, Sebastian Buck, and Bernd Radig. Cooperative probabilistic state estimation for vision-based autonomous mobile robots. In *IEEE Transactions on Robotics and Automation*, volume 18, pages 670–684, October 2002.

[195] B. Schölkopf, C. J. C. Burges, and A. J. Smola. Advances in kernel methodsŮsupport vector learning. In *Cambridge, MA: MIT Press*, 1999.

[196] Bernhard Schölkopf, Sebastian Mika, C. J. C. Burges, P. Knirsch, Klaus-Robert Müller, Gunnar Rätsch, and Alex Smola. Input space versus feature space in kernelbased methods. In *IEEE Trans. Neural Networks*, volume 10, pages 1000–1017, September 1999.

[197] Bernhard Schölkopf and Alex Smola. Support vector machines and kernel algorithms. 2005.

[198] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels.* The MIT Press, 2002.

[199] B. Schott, M. Bajura, J. Czarnaski, J. Flidr, T. Tho, and L. Wang. A modular power-aware microsensor with ¿1000x dynamic power range. Proceeings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005).

[200] David W. Scott. *Multivariate Density Estimation: Theory Practice and Visualization.* NY: John Wiley and Sons, New York, 1992.

[201] Karlton Sequeira and Mohammed J. Zaki. Admit: anomaly-based data mining for intrusions. *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 386–395, 2002.

[202] C. Shannon. Communication in the presence of noise. volume 86. Proceedings of the IEEE, February 2008.

[203] Mohamed A. Sharaf, Jonathan Beaver, Alexandros Labrinidis, and Panos K. Chrysanthis. Tina: A scheme for temporal coherency-aware in-network aggregation. September 2003.

[204] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, June 2004.

[205] Bo Sheng, Qun Li, Weizhen Mao, and Wen Jin. Outlier detection in sensor networks. *Proceedings of the Mobile Ad Hoc Networking and Computing (MobiHoc'07)*, pages 77–82, 2007.

[206] Noritaka Shigei, Hiromi Miyajima, Hiroki Morishita, and Michiharu Maeda. Centralized and distributed clustering methods for energy efficient wireless sensor networks. In *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)*, volume I, Hong Kong, March 2009.

[207] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks.

[208] Edward H. Shortliffe and G. Octo Barnett. *Medical data: their acquisition, storage, and use*, pages 37–69. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[209] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor networkbased countersniper system. Proceedings of the 2nd international Conference on Embedded Networked Sensor Systems (SenSys '04).

[210] Steven W. Smith. The scientist and engineer's guide to digital signal processing, 2nd ed. 1999.

[211] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless Sensor Networks: Technology, Protocols and Applications*. Wiley, 2007.

[212] Mujdat Soyturk, Halil Cicibas, and Omer Unal. *Real Time Data Acquisition in Wireless Sensor Networks*. InTech - Open Access Publisher, November 2010.

[213] C. Spence, L. Parra, and P. Sajda. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. 2001.

[214] Mark Stemm and Randy H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications*, E80-B(8):1125–1131, August 1997.

[215] ICT 2020_4 Scenario Stories. Hidden assumptions and future challenges. 2010.

[216] Diane M. Strong, Yang W. Lee, and Richard Y. Wang. Data quality in context. *Communications of the ACM*, 40:103–110, 1997.

[217] Sharmila Subramaniam, Themis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. Online outlier detection in sensor data using non-parametric models. *Journal of Very Large Data Bases*, pages 187–198, 2006.

[218] Marimuthu Palaniswami James C Bezdek Sutharshan Rajasegarar, Christopher Leckie. Distributed anomaly detection in wireless sensor networks. In *10th IEEE Singapore International Conference on Communication Systems (2006).*

[219] Tsuyoshi Suzuki, Ryuji Sugizaki, Kuniaki Kawabata, Yasushi Hada, and Yoshito Tobe. Autonomous deployment and restoration of sensor network using mobile robots. In *International Journal of Advanced Robotic Systems*, volume 7, pages 105–114, 2010.

[220] P. Sykacek. Equivalent error bars for neural network classifiers trained by bayesian inference. *In: Proceedings of ESANN*, 1997.

[221] He T., Krishnamurthy S., Stankovic J.A., Abdelzaher T., Luo L., Stoleru R., Yan T., and Gu L. Energy-efficient surveillance system using wireless sensor networks. *Mobisys*, June 2004.

[222] Maen Takruri, Subhash Challa, and Ramah Yunis. Data fusion techniques for auto calibration inwireless sensor networks. July 2009.

[223] Pang-Ning Tan. *Knowledge Discovery from Sensor Data*. Sensors Magazine (Cover story), 2006.

[224] Caimu Tang and Cauligi Raghavendra. *Compression techniques for wireless sensor networks*, volume 3. Chapter 10 in book Wireless Sensor Networks, 2004.

[225] L. Tarassenko. Novelty detection for the identification of masses in mammograms. 4, 1995.

[226] Mahbod Tavallaee, Natalia Stakhanova, and Ali A. Ghorbani. Toward credible evaluation of anomaly-based intrusion-detection methods. 40(5):516–524, September 2010.

[227] Giri Kumar Tayi and Donald P. Ballou. Examining data quality. *Communications of the ACM*, 41:54–57, February 1998.

[228] J. R. Taylor. *An introduction to error analysis.*

[229] David J. Thornley, Robert I. Young, and James P. Richardson. From mission specification to quality of information measuresŰclosing the loop in military sensor networks. *In Proceedings of ITA*, 2008.

[230] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman. Taxonomy of wireless micro-sensor network models. *Mobile Computing and Communication Review*, 6, April 2002.

[231] W. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46(2):234–240, 1970.

[232] Daniela Tulone. A resource-efficient time estimation for wireless sensor networks. In *DIALM-POMC'04*, pages 52–59, October 2004.

[233] Daniela Tulone and Samuel Madden. Paq: time series forecasting for approximate query answering in sensor networks. In *EWSN*, pages 21–37, 2006.

[234] V. N. Vapnik. Statistical learning theory. In *New York: Wiley*, 1998.

[235] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Ignacio Soler Jubert, Margaretha Mazura, Mark Harrison, Markus Eisenhauer, and Pat Doody11. Internet of things strategic research roadmap. In *EPoSS*, September 2009.

[236] Agustin Maravall Victor Gómez. *Programs TRAMO (Time series Regression with Arima noise, Missing observations).* Department, Bank of Spain, 1997.

[237] S. V. N. Vishwanathan and Alexander J. Smola. Fast kernels for string and tree matching. In *Advances in Neural Information Processing Systems 15*, pages 569–576. MIT Press, 2003.

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

[239] L. WALD. Some terms of reference in data fusion.

[240] L. Wald. Some terms of reference in data fusion. *In: Geoscience and Remote Sensing, IEEE Transactions on 37.3 (1999)*, page 1190Ű1193.

[241] Douglas D. Walker and Jim C. Loftis. Alternative spatial estimators for ground-water and soil measurements. 35:593–601, 1997.

[242] J. Wallace, D. Pesch, S. Rea, and J. Irvine. Fuzzy logic optimization of mac parameters and sleeping duty-cycles in wireless sensor networks. *In 62nd Vehicular Technology Conference (VTC)*, 3:1824–1828, 2005.

[243] Miaomiao Wang, Jiannong Cao, Jing Li, and Sajal K. Das. Middleware for wireless sensor networks: A survey. *journal of computer science and technology*, 23:305–326, 2008.

[244] Qin Wang, Mark Hempstead, and Woodward Yang. A realistic power consumption model for wireless sensor network devices. In *SECON '06.*, volume 1, pages 286–295, Sept. 2006.

[245] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the First International Conference on Embedded Networked Sensor Systems (Sensys)*, Los Angeles, CA, 2003.

[246] Bing We, Wu Chen, and Xiaoli Ding. Advanced mds based localization algorithm for location based services in wireless sensor network. *IEEE*, 2010.

[247] L. Wei, W. Qian, A. Zhou, W. Jin, and J. X. Yu. Hot: hypergraph-based outlier test for categorical data. 2003.

[248] M. Weiser. The computer for the twenty-first century. In *Scientific American*, pages 94–100, 1991.

[249] Greg Welch and Gary Bishop. An introduction to the kalman filter. *ACM SIGGRAPH International Conference on computer Graphics and Interactive Techniques*, August 2001.

[250] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

[251] Rebecca Willett, Aline Martin, and Robert Nowak. Backcasting: adaptive sampling for sensor networks. In *Proceedings of Information Processing in Sensor Networks*, pages 124–133, 2004.

[252] Alec Woo, Samuel R. Madden, and Ramesh Govindan. Networking support for query processing in sensor networks. June 2004.

[253] Gang Wu, Yi Wu, Long Jiao, Yuan-Fang Wang, and Edward Y. Chang. Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance. *Proceedings of the ACM international Conference on Multimedia*, pages 528–538, 2003.

[254] Wei Wu, Michael J. Black, Yun Gao, M. Serruya, A. Shaikhouni, and Donoghue John P. Neural decoding of cursor motion using a kalman filter. *Neural Information Processing Systems: Natural and Synthetic*, pages 133–140, December 2002.

[255] Weili Wu, Xiuzhen Cheng, Min Ding, Kai Xing, Fang Liu, and Ping Deng. Localized outlying and boundary data detection in sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1145–1157, 2006.

[256] Alex Wun, Milenko Petrovi, and Hans A. Jacobsen. A system for semantic data fusion in sensor networks. *Proceedings of the International Conference on Distributed Event-based Systems (DEBS07)*, pages 75–79, 2007.

[257] F. Xia, Y.C Tian, Y.J. Li, and Y.X. Sun. Wireless sensor/actuator network design for mobile control applications. 7(10), 2007.

[258] N. Xiong and P. Svensson. Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3(2):163–186, 2002.

[259] J. Xu and X. Tang.

[260] Chin-Lung Yang, S. Bagchi, and W. J. Chappell. Location tracking with directional antennas in wireless sensor networks. *IEEE MTT-S International Microwave Symposium Digest*, 2005.

[261] Xingwei Yang, Longin Jan Latecki, and Dragoljub Pokrajac. Outlier detection with globally optimal exemplar-based gmm. In *SIAM International Conference on Data Mining*, pages 145–154, 2009.

[262] Shuncai Yao, Jindong Tan, and Hongxia Pan. A sensing and robot navigation of hybrid sensor network. In *Wireless Sensor Network*, pages 267–273, 2010.

[263] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Analysis of a prediction-based mobility adaptive tracking algorithm. Boston, 2005. Proceedings of the IEEE Second International Conference on Broadband Networks (BROADNETS).

[264] Ossama Younis and Sonia Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. *Mobile Computing, IEEE Transactions*, 3(4):366–379, 2004.

[265] Yingqin Yuan and Moshe Kam. Distributed decision fusion with a random-access channel for sensor network applications. In *IEEE Trans. Instr. Meas.*, volume 53, pages 1339–1344, August 2004.

[266] M. Yusuf and T. Haider. Energy-aware fuzzy routing for wireless sensor networks. *In IEEE International Conference on Emerging Technologies (ICET'05)*, pages 63–69, 2005.

[267] Sadaf Zahedi and Chatschik Bisdikian. A framework for qoi-inspired analysis for sensor network deployment planning. In *Proceedings of the 3rd international conference on Wireless internet*, WICON '07, pages 28:1–28:8, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[268] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. 2003.

[269] Yang Zhang, Nirvana Meratnia, and Paul Havinga. Outlier detection techniques for wireless sensor networks: A survey. *IEEE communications surveys and tutorials*, 12(2), 2010.

[270] Feng Zhao and Leonidas Guibas. *Wireless Sensor Networks – An Information Processing Approach.* Elsevier / Morgan-Kaufman, Amsterdam, 2004.

[271] Feng Zhao, Jie Liu, Juan Liu, Leonidas Guibas, and James Reich. Collaborative signal and information processing: An information directed approach. In *Proceedings of the IEEE*, volume 91, pages 1199–1209, August 2003.

[272] Zibin Zheng, Jian Wanga, and Ziyu Zhu. A general anomaly detection framework for internet of things. June 2011.

[273] Yongzhen Zhuang and Lei Chen. In-network outlier cleaning for data collection in sensor networks. In *In CleanDB, Workshop in VLDB*, pages 41–48, 2006.

[274] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering support vector machine kernels that recognize translation initiation sites. 16, 2000.