UNIVERSITÀ DEGLI STUDI DI PALERMO
Dipartimento di Ingegneria Chimica, Gestionale, Informatica, Meccanica
Dottorato di Ricerca in Ingegneria Informatica

A PROBABILISTIC APPROACH TO ANOMALY DETECTION FOR WIRELESS SENSOR NETWORKS

SSD: ING-INF/05

TESI DI
ING. ALFONSO FARRUGGIA

COORDINATORE DEL DOTTORATO
PROF. SALVATORE GAGLIO

TUTOR
PROF. GIUSEPPE LO RE

CICLO XXIII

DOTTORATO

This work is dedicated to
the my grandfather Andrea.

Sommario

Negli ultimi anni, l'Ambient Intelligence (AmI) ha suscitato un crescente interesse nelle comunità scientifica ed industriali. L'intelligenza artificiale e la domotica di ultima generazione si sposano nell'AmI e ne fanno una delle discipline più apprezzate nel mondo della ricerca. In questo nuovo paradigma gli ambienti della vita quotidiana sono provvisti di dispositivi elettronici capaci di interagire con gli utenti, al fine di soddisfarne i loro bisogni. I dispositivi elettronici di un sistema per l'AmI hanno caratteristiche stringenti: sono pervasivi, piccoli nelle dimensioni e soprattutto *intelligenti.* Infatti, essi permettono di acquisire ed elaborare dati ambientali, in modo da creare comfort per l'utente. Una tecnologia abilitante per l'AmI è rappresentata dalle reti di sensori wireless (WSN), in questi anni, le WSN hanno guadagnato popolarità come strumento per il monitoraggio ambientale, soprattutto grazie alla loro semplice modalità di utilizzo, al basso costo, e alla loro non intrusività. Un potenziale svantaggio delle WSN è quello di garantire la costante affidabilità del loro funzionamento nel tempo.

Questa tesi presenta un algoritmo che permette la rilevazione dei nodi di una WSN affetti da anomalie. Per stimare lo stato di salute di un nodo, il presente approccio sfrutta solo i dati acquisiti dalla WSN, integrando anche informazioni su possibili fattori di disturbo. In particolare, il metodo presentato sfrutta un approccio probabilistico, combinando due tipi di modelli grafici: Markov Random Fields e le Bayesian Networks. Il primo strumento è utilizzato per effettuare una stima, a volte imprecisa, dello stato di funzionamento del sensore presente su un nodo, mentre la seconda, raffina la stima precedente, inferendo lo stato generale di salute del nodo. L'approccio proposto è stato testato su un set di dati reali acquisiti in un ambiente di lavoro, caratterizzato dalla presenza di attuatori che perturbano il reale andamento delle grandezze fisiche monitorate. Le prestazioni del presente approccio sono state valutate secondo le classiche metriche statistiche per la valutazione dei classificatori binari apportando risultati soddisfacenti durante gli esperimenti.

Abstract

Ambient Intelligence (AmI), as a specific discipline, has gained increasing attention from the academic and industrial communities in the past few years. Its distinctive feature is the focus on the connection between Artificial Intelligence and the latest generation of Home Automation. The aim of this new paradigm is to augment everyday life environments where electronic devices have been deployed with the capability of interacting with the users in order to satisfy their needs.

The electronic devices of an AmI system need to be pervasive, small and smart; they are assumed to acquire and process environmental data in order to create comfort for the user. An enabling technology for AmI is represented by Wireless Sensor Networks (WSNs), which are also gaining popularity as a tool for environmental monitoring, especially thanks to their use of simple, inexpensive, off-the shelf sensors and to their unintrusive deployment. A potential drawback regards the fact that ensuring the reliability of their operation is not trivial, and faulty sensors are not uncommon. Moreover, the deployment environment may influence the correct functioning of a sensor node, which might thus be mistakenly classified as malfunctioning.

The topic of this thesis is the proposal of algorithms for detecting faulty WSN nodes by analyzing the data acquired from each of their on-board sensors, and integrating relevant information about potential environmental influence.

In particular the presented method uses a probabilist approach to estimate the health status of node based on the combination of two types of Graphical Models, namely: Markov Random Fields and Bayesian Networks.

The former is used to estimate the raw status of a sensor, whereas the latter refines the previous estimation inferring the actual overall health status of node.

The proposed approach was tested on a real dataset acquired in a work environment characterized by the presence of actuators, that also affect the real trend of the monitored physical quantities. The method showed good performance in terms of well-known metrics to evaluate the binary classifiers.

Acknowledgments

I would first like to thank my advisors, Prof. Giuseppe Lo Re and Marco Ortolani. Prof. Giuseppe Lo Re encouraged me to follow this interesting research direction, and for his important advices for my work. Marco Ortolani for his patience and for the infinite time spent to our brainstorming. My family deserves a precious thanks for all the support they have given me during these three years of PhD. I would to thank my colleagues Daniele and Liliana for them scientific debates in front of the coffee machine; at the end I would to thank all others my friends to their encouragement during this period of my life, specially Angelo Fabio, Gianluca e Rossella.

Contents

Abstract **iii**

Acknowledgments **iv**

Glossary **xi**

1 Introduction **1**

1.1 Motivations and Goals 1

1.2 Contributions 8

1.3 Dissertation Outline 9

1.4 Publications 9

2 Scientific Background **11**

2.1 Fault Detection Systems 11

2.2 Graphical Models 14

2.2.1 Markov Random Fields 14

2.2.2 Bayesian Network 21

3 An Architecture for Anomaly Detection **24**

3.1 The Logical Organization of the System 24

3.1.1 Physical Layer 25

3.1.2 Estimation Layer 26

3.2 Health sensor estimation 27

3.2.1 Inferring the health status of sensor 28

3.3 Node Health Estimation 33

3.3.1 Specializing the Model for Indoor Environmental Monitoring 34

3.3.2 Inferring the Health Status of an Environmental Sensor Node 37

4 UML Modeling 39

4.1 UML Project 39

4.1.1 Deployment Diagram 40

4.1.2 Software Component Diagram 43

4.1.3 Class Diagram of System 44

4.1.4 State Chart Diagram of System 45

4.2 ER-Diagram 46

5 Experimental Results 51

5.1 MRF experiments 51

5.1.1 Scenario 1: Dataset influenced by Continuous errors . . . 54

5.1.2 Scenario 2: Dataset influenced by Gaussian errors 55

5.1.3 Scenario 3: Dataset influenced by discontinuous errors . 56

5.2 BN experiments 57

5.2.1 Scenario 1: Dataset influenced by actuators 60

5.2.2 Scenario 2: Dataset influenced by a simulated faulty . . . 61

5.2.3 Scenario 3: Dataset influenced by actuators and a simulated error 63

6 Conclusions 66

Bibliography 68

List of Figures

1.1 Comparison between the structures of the human retina and the proposed WSN. 3

1.2 The human language comprehension model vs the proposed hierarchical reasoning model. 4

2.1 Neighborhood system 16

2.2 First order Clique 17

2.3 Second order Clique 18

2.4 Peaking control with temperature 18

2.5 Sample Bayesian Network, showing hidden and observed nodes. 22

2.6 A Bayesian network, highlighting message passing between two showing hidden and observed nodes. 22

3.1 Architecture of proposed work, two main layer: PHYSICAL LAYER to acquire environmental information, ESTIMATION LAYER to process the data acquired 25

3.2 Environmental data and status of actuators are gathered and send to central server. 26

3.3 Representation of a group of sensors as nodes of a MRF; the health status of each sensor s_i is represented by an observable variable y_i, and a hidden variable x_i. 28

3.4 Clique is highlighted by gray area, red line indicates the cut-off distance to discovery the subset of nodes. 29

3.5 The proposed Bayesian network. 34

3.6 Impact of various factors on the functioning of the node: influence of the actuator on temperature 35

3.7 Impact of various factors on the functioning of the node: influence of the battery charge. 36
3.8 A Bayesian network, highlighting message passing between two showing hidden and observed nodes. 37

4.1 Component Diagram for Hardware 41
4.2 Software Component Diagram 42
4.3 Class Diagram 48
4.4 State Chart Diagram 49
4.5 ER-Diagram 50

5.1 Map of the sensor field from Intel Berkeley Research Lab [1], highlighting regions of correlated sensor readings. Red line reads the status of the artificial light, green line reads the status of the air conditioner, whereas the blue line reads the status of window. Sensor nodes capture environmental information and send them to database. 52
5.2 Cliques graph for Berkeley Laboratory 53
5.3 Sample of the real dataset, with 5% of the nodes corrupted by *continuous* constant fault. 55
5.4 Plots showing how the algorithm detects the health status for a faulty sensor. 56
5.5 Sample of the real dataset, with 5% of the nodes corrupted by *discontinuous* fault. 57
5.6 Plots showing how the algorithm detects the health status for a faulty sensor. 58
5.7 A WSN deployed in an office environment, with 5 sensor nodes and one actuator (the air conditioner, AC) 60
5.8 Environmental information accounted in the Bayesian classifier, the errors of classification are committed by the classifiers MRF based 61
5.9 Real dataset of temperature perturbed by a Gaussian error . . . 62
5.10 Real dataset of humidity perturbed by a Gaussian error 63

5.11 Progress of belief of the node 5 during the errors occurred in its sensors, the last three charts indicate the period which the error occur respectively on the sensor of temperature, humidity, and light. 64
5.12 Real dataset of humidity perturbed by the air conditioner and by a fault. 65
5.13 Dynamics of the estimate of the status for the classifiers in scenario 3. 65

List of Tables

5.1 Continuous Faults 54

5.2 Discontinuous Faults 59

5.3 The sensors used for environmental monitoring, and their characteristics. 59

5.4 Performance summary of the experimental scenarios. 64

Glossary

AmI	Ambient Intelligence
BN	Bayesian Network
BP	Belief Propagation
CSFD	Collaborative Sensor-Fault Detection
GM	Graphical Model
GRF	Gibbs Random Field
ICM	Iterated Conditional Modes
MRF	Markov Random Field
OMG	Oriented Management Group
PCA	Principal Component Analysis
SA	Simulated Annealing
TTL	Transistor-Transistor Logic
UML	Unified Modeling Language
WSN	Wireless Sensor Network

Chapter 1

Introduction

Anomaly detection systems allow to detect non conformant behaviors in automation systems, and they are used in many applications such as fraud detection for credit cards, insurance or health care, intrusion detection for cyber-security, fault detection in safety critical systems, and military surveillance for enemy activities. An anomaly is the deviation from the normal behavior of a process and can be attributed to several causes. A system affected by anomaly could cause damages to itself or to the user.

This work presents an anomaly detection system developed for a Wireless Sensor Network (WSN) and, in particular, it is customized for an Ambient Intelligent (AmI) scenario. Avoiding anomalies in an automation system means making it reliable, which in an AmI scenario ensures the user's wellness. In order to realize an anomaly detection system many techniques have been developed, most of these are based on Machine Learning and Artificial Intelligence methods. In particular, to carry out this work two efficient techniques, the Markov Random Field (MRF) and the Bayesian Network (BN) have been adapted to the considered scenario.

1.1 Motivations and Goals

For several decades the influence of automation systems on daily lives has grown exponentially producing an increase of the demand for more efficient and higher quality systems and products. In recent years, industries of sophisticated automation systems and artificial intelligence discipline have created the concept of Ambient Intelligence [2, 3, 4]. In this context the user is the center of a pervasive digital intelligent environment, whose primary goal consists in satisfying users' requirements as regards controlling the conditions of their surroundings. Thanks to continuous progress in microelectronics, devices with different capabilities and user interfaces are increasingly becoming part of our daily life. These technologies can be interconnected and managed using intelligent software able to understand situations and events relevant to make decisions in order to modify some aspects of our life.

Nowadays small chips are installed into many devices used in our daily lives. This permits to develop devices that acquire information from environment in order to obtain knowledge about the users' lifestyle.

For instance, considering a home automation system, these devices may be installed in a house to turn on or off the lights in anticipation of the user's actions, or, in an elderly care scenario to book a room at the hospital; hence, it is obvious that these devices are changing our life style increasing our wellness.

Such computing devices will be coordinated by intelligent systems to integrate the various resources available and provide an intelligent environment. The confluence of several disciplines has led to the introduction of the new research field called Ambient Intelligence (AmI), whose purpose is to create a digital environment that proactively and appropriately supports people in their daily lives.

An innovative AmI architecture is inspired by the human nervous system, in which signals gathered by the peripheral system are filtered, aggregated and then sent to the central system for high-level processing.

A remarkable example is the processing of visual information occurring in the retina [5]: in the human eye, photoreceptors convert light into electrical signals that are passed to a network of retinal neurons, and are modified before being transmitted to gangliar neurons; eventually, they are handed to the optic nerve that carries the information up to the brain. The retinal neuron network does not restrict itself to carrying signals from photoreceptors, but rather combines them to obtain an aggregate heavily dependent on the spatial and temporal features of the original light signal.

In an AmI architecture the terminal sensory component performing is rep-

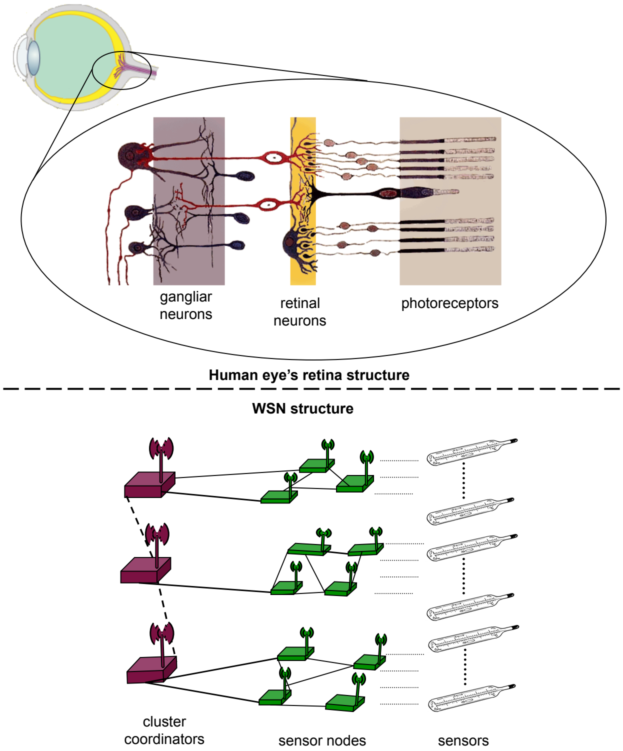

Figure 1.1: Comparison between the structures of the human retina and the proposed WSN.

resented by smart devices pervasively deployed in the environment. Figure 1.1, partially drawn from [6, 7], highlights the similarity between the structures of the human visual organ and intelligent devices employed here.

A further example is shown in [8]; a clustered network structure is proposed in which each small cluster, constituted by heterogeneous devices with different computational capabilities, distributedly processes homogeneous data. This pre-processing phase exploits spatio-temporal correlation of data, in order to compute a model that nodes will share, thanks to their cluster coordinator, similarly to the approach proposed in [9].

The AmI system is organized according to a hierarchical structure whose modules are combined together in order to carry on specific reasoning on the environment at different levels of abstraction and on different kinds of perceptions. The overall behavior mimics that of the human brain, where the

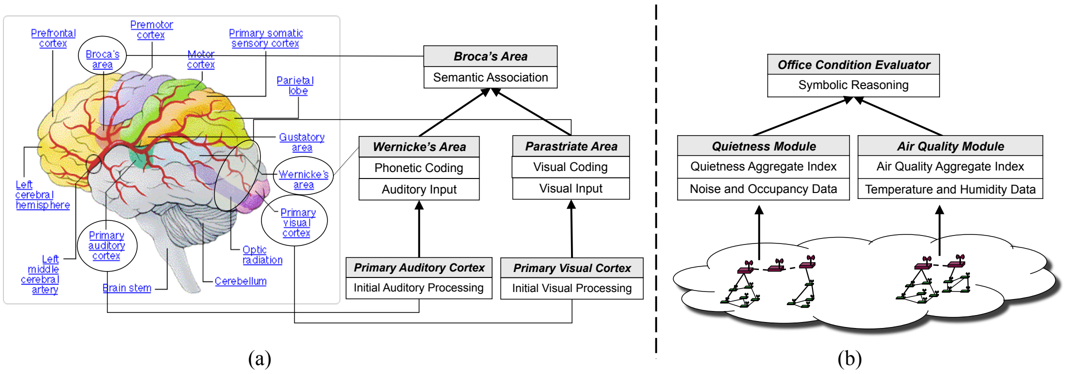

Figure 1.2: The human language comprehension model vs the proposed hierarchical reasoning model.

emerging complex behavior is the result of the interaction among smaller subsystems. From the design point of view, the modular organization allows for the realization of a scalable software architecture, able to effectively manage the huge amount of sensory data.

Figure 1.2, partially taken from [10], draws a parallel between the human brain model and the system proposed in [8] . In this architecture, the outcome of lower-level reasoning is fed into the upper levels, that deal with the integration of information originated by multiple lower-level modules. Each module independently measures environmental quantities, conceptualizes them, and describes the extracted concepts linguistically. Moreover various modules process both direct and indirect measurements; the former occur at modules located at the lowest level in the hierarchy, while the latter are carried on at the upper layers, mediated by their lower-layer counterparts.

Another particular scenario that shows the analogy between the AmI system and the human brain can be the human language comprehension model, described in [5]. This provides a significant example of interaction patterns among specific areas of the brain, as schematically presented in the left side of Figure 1.2. Different anatomic structures are devoted to different phases of language processing: the primary auditory cortex initially processes the auditory signals while at the same time the primary visual cortex processes the visual signals. Pieces of information separately obtained by each low-level structure are sent to the areas devoted to phonetic and visual coding respectively. The outcome of the two intermediate modules are passed to the semantic association area, where they are merged.

In order to clarify the importance of AmI in the context of everyday life, an example which has been reported many times in literature may be provided [11]:

> “Ellen returns home after a long day’s work. At the front door she is recognized by an intelligent surveillance camera, the door alarm is switched off, and the door unlocks and opens. When she enters the hall the house map indicates that her husband Peter is at an art fair in Paris, and that her daughter Charlotte is in the children’s playroom, where she is playing with an interactive screen. The remote children surveillance service is notified that she is at home, and subsequently the on-line connection is switched off. When she enters the kitchen the family memo frame lights up to indicate that there are new messages. The shopping list that has been composed needs confirmation before it is sent to the supermarket for delivery. There is also a message notifying that the home information system has found new information on the semantic Web about economic holiday cottages with sea sight in Spain. She briefly connects to the playroom to say hello to Charlotte, and her video picture automatically appears on the flat screen that is currently used by Charlotte. Next, she connects to Peter at the art fair in Paris. He shows her through his contact lens camera some of the sculptures he intends to buy, and she confirms his choice. In the mean time she selects one of the displayed menus that indicate what can be prepared with the food that is currently available from the pantry and the refrigerator. Next, she switches to the video on demand channel to watch the latest news program. Through the follow me she switches over to the flat screen in the bedroom where she is going to have her personalized workout session. Later that evening, after Peter has returned home, they are chatting with a friend in the living room with their personalized ambient lighting switched on. They watch the virtual presenter that informs them about the programs and the information that have been recorded by the home storage server earlier that day”.

The example shows that in Ellen’s house there are several kinds of sensors which interact with each other and with Ellen. To realize the system presented in the example the best enabling technology is represented by Wireless Sensor

Networks (WSNs) [12, 13], thanks to their capacity of providing a pervasive and unintrusive means for sensing the environment.

Wireless Sensor Networks (WSN) are nowadays increasingly gaining popularity, also in challenging scenarios such as pollution control, intrusion detection, healthcare monitoring [14]. In particular, the sensor nodes are equipped with one or more sensors capable of acquiring different physical environment, a radio component, a microprocessor. The WSN is used primarily to acquire data in hostile environments, in fact, the sensor nodes are designed to communicate by radio and their are powered by a battery that ensures a certain charge in time, in some cases can be equipped with systems acquisition of electricity, for example the photovoltaic panel. In a WSN, one or more gateway nodes may be have the task of collecting data from the remaining nodes of the network and send them to a processing station, exploiting the Internet. These devices have more computing power, more availability of electricity with respect to the sensor node and typically are located in places more accessible to person. The applications for WSNs can be classified into two categories, for monitoring and tracking. The former provide monitoring of environments both indoor and outdoor, scenarios for health care, for the processes automation, seismic monitoring, etc. Applications for tracking instead provide for the tracking of objects, animals, humans and vehicles.

WSNs are used to acquire information related to the environment and to the users that live therein. A WSN is reliable if gathered data are not corrupted, which in turn results in reliable extracted knowledge to be used by the AmI system to make the environment comfortable for the users.

In order for the sensor network to function properly in the real world, it is important to provide an anomaly detection system for sensor nodes to isolate faults which could mislead the data acquired by network.

Generally, complex automation systems, like AmI systems, need sophisticated management to guarantee the working correctness. A challenge of modern automation systems is the automatic supervision that introduces new research fields and corresponding theoretical models.

Initially the monitoring systems were achieved by using the *check limit* technique, and by monitoring some important variables of the system. The protection mechanisms are in this case started by human operator only if the alarm has been triggered, i.e. if the monitoring variable has exceeded a pre-set threshold.

The *check limit* technique is totally inadequate to cope with increasing complexity of automation systems; for instance, sometimes just the mere triggering of an alarm, may by itself, causes irreversible damage to the system. Nowadays, modern artificial intelligence algorithms show that the faults in system automation can be prevented by avoiding greater damage both in terms of costs and with respect to the people who use its features. The primary objective of these methods is to implement early detection of anomalies in actuators, processes, components and sensors. In [15], the authors classified typical anomalies in datasets into three categories, namely *point*, *contextual*, and *collective* anomalies. Considering a dataset of such physical quantities, a single reading can be regarded as a *point* anomaly if it does not fit with the rest of the data, as is the case, for instance, for outliers; an anomaly must be classified as *contextual* whenever the structure of nearby data is to be taken into account in order to recognize it; such context needs to be specified as a part of the problem formulation. Finally, *collective* anomalies are indistinguishable from the expected data, and may only be identified by considering the overall dataset.

The main goal of this thesis is to present a probabilistic method to detect anomalies in wireless sensor nodes, in particular the kind of anomalies to be discovered fall into the *contextual* category. The approach estimates the health of the sensor node by processing only the data acquired from sensors installed on board, and also accounting for the environmental information. It will be shown that the proposed system is able to help prevent anomalies in WSN analyzing only the data gathered by nodes. To guarantee that an AmI system is free from anomalies corresponds to increase its reliability, so that for instance other dangerous actions are not performed, since a system affected by anomalies could expose to danger both the user's life and the environment or could be cause of economic loss.

For these reasons an anomaly detection system is considered a fundamental part of an AmI system and, in general, of a complex automation system. This thesis realizes anomaly detection based on machine learning techniques, in particularly Markov Random Fields and Bayesian Networks. More specifically, to analyze a node of a WSN, each of its embedded sensors is assessed, and the employed method is the MRF. This technique permits to exploit spatial information in a classification process. In more complex scenarios, like those for AmI, the spatial correlation between sensors is sometimes broken because some external factors influence the data gathered by the sensors. To avoid this problem, the external factors are modeled and these informations are used in a BN in order to better estimate the health status of the whole node.

1.2 Contributions

This work presents an approach to anomaly detection for WSN. The approach is structured as a multilayer architecture. In particular it presents a structure constituted by two layers: the former implements the physical architecture to acquire environmental data, whereas the latter implements the reasoner to infer the health status of a node.

To implement this system, appropriate machine learning tools have been used. The main contribution of this thesis consists in the adaptation of the original machine learning tools in order to form a chain for knowledge extraction with the aim of inferring the overall health status of a sensor node.

Initially the data are manipulated by a Markov Random Field; computing the energy of the field permits to infer which sensor installed on board of a WSN node is probably damaged. The sensor considered as an outlier (i.e. affected by an anomaly) is detected because it is compared, using the MRF, with its neighbors using only the acquired data. This concept exploits the property of spatial statistical correlation of neighbors so that neighbors nodes acquire similar data.

On board of a node are installed different sensor to acquire some physical quantities. If a node presents a damaged sensor it is not correct to infer that the entire node is damaged. The next step of the present work is to infer the overall status of a WSN node, which is realized by joining the information computed via MRF for all sensors thanks to the use of a Bayesian Network. In this phase, environmental information that could alter the data gathered is accounted for.

The experiments proving the effectiveness of the work have been performed using real datasets, and the designed algorithms developed using Matlab, in simulation, and Java, for real-world experiments; the final tool is presented in Chapter 4.

1.3 Dissertation Outline

The remainder of the dissertation is organized as follows.

Chapter 2 describes mathematical tools exploited to implement the proposed approach. In particular it describes the Markov Random Field (MRF) methodology focusing the Ising model to compute the energy of the field, and it gives an overview on Bayesian Network (BN) explaining the Belief Propagation (BP) approach to infer the BN variables.

Chapter 3 describes the proposed architecture showing as mathematical tools were used to implement the architecture. It shows the sub-parts of the proposed architecture and how these parts interact to infer the status of a sensor node.

Chapter 4 presents the software tool implementing the proposed approach. The software is designed according to UML standard, and it is developed in Java.

Chapter 5 describes the experimental results. Various application scenarios have been deployed to prove the effectiveness of the proposed work, which was assessed by measuring such metrics as accuracy and precision.

Finally, Chapter 6 reports the conclusion about this work, and states some possible future directions for research.

1.4 Publications

- A. De Paola, A. Farruggia, S. Gaglio, G. Lo Re, and M. Ortolani, "Exploiting the human factor in a wsn-based system for ambient intelligence," in *Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. International Conference on*, pp. 748–753, IEEE, 2009.
- A. Farruggia, M. Ortolani, and G. Lo Re, "FDAE: A failure detector for asynchronous events," in *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*, pp. 197–202, IEEE, 2010.
- A. Farruggia, G. Lo Re, and M. Ortolani, "Detecting faulty wireless sensor nodes through stochastic classification," in *Pervasive Computing and Communications Workshops (PERCOM Workshops),*

2011 IEEE International Conference on, pp. 148–153, IEEE, 2011.

- A. Farruggia, G. Lo Re, and M. Ortolani, “Probabilistic anomaly detection for wireless sensor networks,” in *Proceedings of the 12th international conference on Artificial intelligence around man and beyond*, AI*IA’11, pp. 438–444, Springer-Verlag, 2011.

Chapter 2

Scientific Background

This chapter shows the related works and mathematical tools exploited to implement the proposed approach. Anomaly detection is a largely examined research area. Many systems in real application use WSNs as a tool to interact with environment or people; it is thus intuitively evident that if the underlying WSN is affected by an anomaly, the whole system will be malfunctioning.

In literature many works use the WSN itself as a tool to discover anomalies, whereas the focus of the present thesis is the detection of anomalies *within* the WSN. The following Sections present the theory of mathematical tool used to realize this work; in particular two machine learning tools have been used which belong to the family of Graphical Models, namely: the Markov Random Field (MRF) methodology focusing on the Ising model to compute the energy of the field, and the Bayesian Network (BN) explaining the Belief Propagation (BP) approach to infer the BN variables.

2.1 Fault Detection Systems

The topic of anomaly detection for Wireless Sensor Networks has already been addressed, and remarkable results are reported in literature; however, many works [16, 17, 18, 19] fail to consider peculiar characteristics that may negatively affect the estimate of the health status of a sensor node. The purpose of the present work is specifically focused on monitoring wireless sensor nodes in order to provide anomaly detection. The anomaly detectors presented in literature fall into two main categories: those that rely on some other external control mechanism, as presented in [20], and the ones directly exploiting the sensed data in order to assess the status of the monitored element.

For the former approach many works are examined, for example [21] presents a failure detector based on a handshake mechanism evaluating the delays of messages exchanged between the monitored and the managed element, another approach presented in [22] exploits an query/ACK mechanism, which however requires a reliable timeout estimate in order to properly set the monitoring interval. To this purpose this failure detector uses the history of past estimates to compute new values for both quantities.

In the present work, the latter approach is adopted, which is aimed to assess the health status of a sensor node exclusively by analyzing the sensor readings; in detail, it will use a probabilistic approach to estimate the status of monitored element. Similar approaches have been presented in literature, but they have a focus on different application scenarios.

For instance, in [23] a stochastic recursive identification algorithm is presented which can be implemented in a fully distributed and scalable manner within the network. The authors demonstrate that it consumes modest resources as compared to a centralized estimator, while still being stable, unbiased, and asymptotically efficient; in the considered scenario, sensors classify the presence or absence of an effluent released from a chemical plant into a river, or in [24], the authors present a failure detector based on data mining, able to detect faults in an electric power system.

Focusing on the Wireless Sensor Network, in literature relevant works are presented that implement a fault detection, but all of these present some aspects that can be improved. For example, the authors of [25] present an efficient collaborative sensor-fault detection (CSFD) scheme, where the health status of a sensor node is inferred via a homogeneity test. Similarly of the present work, CSFD implements a probabilistic approach, although it relies on specific control messages thus causing additional overhead, not required by presented approach.

In [16], the authors present an approach for identifying regions of faulty sensor nodes, and show that the probability of a correct diagnosis is satisfactory for large faulty sets. They distinguish among faults occurring at different layers of the sensor network, such as the physical layer, hardware, system software, and middleware; however, they later focus only on hardware level malfunctioning. The proposed algorithm estimates the probability that a node is faulty valuing the elapsed time of a test. The present approach, on the other hand, does not depend on the specific layer where the faults occur; moreover, the probability of a correct diagnosis is independent of the amount of faulty nodes.

The authors of [17] present a small-scale WSN; in particular, a method for detecting faulty sensor nodes using Principal Component Analysis (PCA) and wavelet decomposition is showed. The PCA-based model is computed by analyzing historical data, and sensor readings are decomposed into a principal space and a residual subspace. The authors claim that faulty sensor nodes can be detected by extracting high-frequency coefficients of wavelet decomposition, although the potential impact of an external factor, such as the presence of actuators, on a healthy node is not considered, which may alter the outcome of the proposed algorithm.

The authors of [18] present a probabilistic fault detection algorithm to select the sensor nodes to be used as probe stations; the algorithm takes into account the probability distribution of sensor nodes, and the probability distribution function for faults in wireless sensor networks. The node chosen as a probe node is the one with the lowest probability of being faulty, which allows it to be assigned additional control tasks, such as sending probing packets, receiving feedback messages and sending control messages. In presented work, on the other hand, does not require the presence of special control nodes; all nodes are assumed to show the same behavior, and have the same probability of being faulty; moreover, no additional control overhead is required.

Markov Random Fields and Bayesian Networks are the mathematical tools exploited in this work, in literature others works use them, now some of these works are presented. In particular for the Markov Random Fields, the work in [26] present a framework for distributed signal processing in sensor network environments, in which sensor nodes collect noisy readings, and classify them by using a Markov Random Fields; unlike this presented approach, however, the authors propose a static calibration of the necessary MRF parameters, instead, in [27] MRFs are used to identify the most relevant collected data, in order to implement an algorithm for aggregation of large amounts of data originating from diverse sources; unlike this work, however, the possibility of anomalies introduced by faulty sensors is not taken into account.

Instead, in [19], the authors address potential errors in sensor measurements due to faults, and develop a distributed Bayesian algorithm for detecting and correcting such faults. They present a sample scenario where sensors are deployed in a region where concentrations of some chemical agent may exceed some pre-defined threshold, and propose a Bayesian approach based on the assumption that measurement errors due to faulty equipment are likely to be uncorrelated, whereas environmental measurements are spatially correlated. A similar assumption is used in this approach, although the application domain is intrinsically different, and presented approach is applicable to generic off-the-shelf sensors, and takes the specific operational context into account.

2.2 Graphical Models

Graphical Models merge the probability theory with the graph theory. They permit to model problems of real scenarios using a simple graphical visualization and resolving them through probabilistic approach. In Graphical Models each node in the graph represents a random variable, whereas the edges, which connect the variables, represents the relations between them. The graphical model can belong to two main category according to the graph is directed or undirected. In the former each node is a random variable and the edges represent statistical dependencies between the variables, whereas in the latter the joint probability over all variables can be written in a factored form. In this work two different approach of graphical models are used, in particular for the undirected graph it is used the Markov Random Field, whereas for the direct graph bayesian network is used. To understand these methods the main theory is presented below.

2.2.1 Markov Random Fields

MRFs are a mathematical tool that allows to exploit spatial information in a classification process, where the considered stochastic variables are assumed to have Markov properties, and have been widely used in the classification of data from spatial databases [28]. MRFs allow to reduce a global model of a wide dataset into an equivalent model based only on the local properties of data.

Features of Markov Random Field

In probability theory, stochastic process has the Markov property if the random variable of future states of the process, given the present state, depends only on the present state. Variables that have this property form the simplest Markov model that is the Markov chain.

Markov Chain

A random process $X = (X_n)_{n=1}^{N}$, with $X_n \in \Lambda$, is a 1^{st} order Markov chain if:

$$p(x_n|x_{n-1}, \dots, x_1) = p(x_n|x_{n-1}) \forall n > 1.$$

The chain is said to be homogeneous if the transition probabilities

$$p(X_n = i|X_{n-1} = j) = f(i, j)$$

do not depend on n.

Markov chain is defined through its priori distribution $P =_{def} p(x_1)$, and its transition matrix $M(f(i, j))_{i,j\in\Lambda}$. Considering a scenario with multidimensional processes, the Markov chain is extends in Markov field where the variables exchange information on lattice structure. Each variable interact with variables around, for the Markov property its value is independent of the rest of the Markov field. To give a basic definition of MRF approach first proceeding to explain the concepts of Neighborhood System, after the Random Field concept is explained and finally the Ising Model to compute the energy of the Field.

Neighborhood System

To give a basic definition of Neighborhood System, let us to consider a lattice $S \equiv S_1, \dots, S_N$ of finite dimension N. A neighborhood system n on S is defined as a collection of subset n_s of S,

$$n \equiv \{n_s : s \in S, n_s \subset n\}$$

where for each n_s, neighborhood of site s, holds

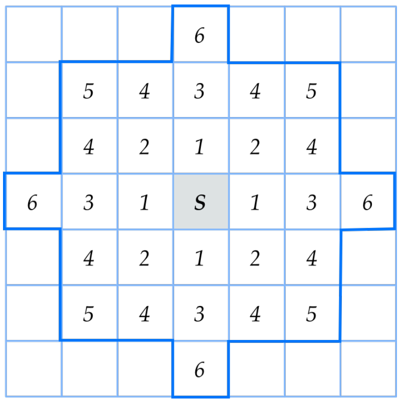

Figure 2.1: Neighborhood system

1. s does not belong to s_n

2. $r \in n_s \implies s \in n_r, \forall s \in S$.

In figure 2.1 examples of neighborhood systems are showed. Usually a neighborhood systems is showed using the notation n^m where the neighborhood is said of order m. For example neighborhood $n^1 = \{n_s^1\}$ is the set of the 4 closest sites, except those on the border, n_s^2 takes the 8 closest neighbors, and so on.

Clique

A subset $c \subseteq S$ is a clique with respect to n if one of the following conditions is satisfied:

1. c is a sigle site

2. every pair (r, s) of distinct sites in c are neighbors, that is:

$$r \neq s \implies r \in n_s$$

C=$C(S, n)$ denotes the set of cliques with respect to S and n. In figure 2.2 and in figure 2.3 all possible cliques corresponding to systems n_1 and n_2 are shown.

Random Field

A random field (RF) defined on a lattice S is a set of random variables $X = \{X_s\}, \forall s \in S$. The $\Omega = \Lambda^N$ is the space of the possible value x that a random variable X can assume,

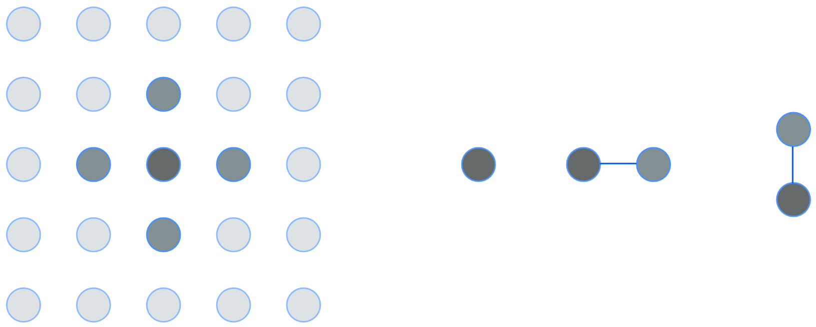

Figure 2.2: First order Clique

$$\{X = x\} \iff \{X_1, = x_1, \ldots, X_N, = x_N\} \forall x \in \Omega$$

where Λ is the space of a single variable x_s.

Definition 1 (Markov Random Field). *A random field X defined on a lattice S in a MRF with respect to a neighborhood system n if [29, 30]*

1. $p(x) > 0 \forall x \in \Omega$

2. $p(x_s|x_r, r \in S, r \neq s) = p(x_s|x_r, r \in n_s)$,

for every $s \in S$ and $x \in \Omega$.

The function $p(x_s|x_r, r \in n_s)$ is called the local characteristics of MRF. According to [31], for each process the joint probability p(x) is determined by these conditional probabilities.

Each X is a MRF if satisfies (1) and the neighborhoods are large enough to encompass all existing dependencies, this could produce a big computational load. To avoid it a MRF can be written as a Gibbs Distribution, and it better explained below.

Gibbs Distribution/Gibbs Random Field

Considering the pair $\{S, n\}$, the Gibbs distribution is a probability P on Ω represented as follow [30]:

$$P(x) =_{def} p(X = x) = \frac{1}{Z} \exp[-\frac{U(x)}{T}]$$

where Z and T are constants and U, called the energy function, has the form

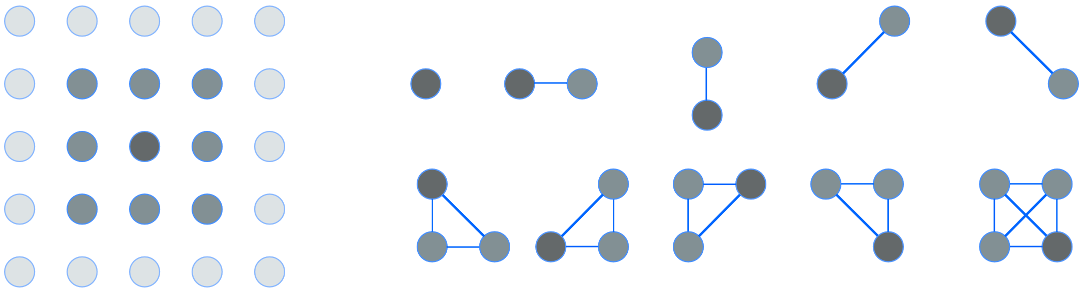

Figure 2.3: Second order Clique

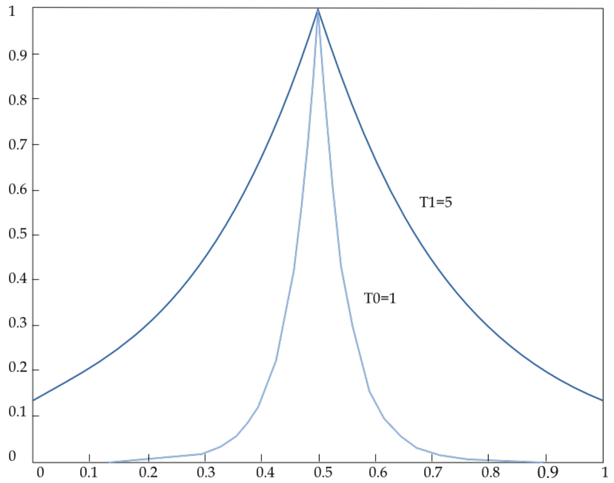

Figure 2.4: Peaking control with temperature

$$U(x) =_{def} \sum_{c \in C} V_c(x)$$

Considering C as the set of cliques for n, V_c is a function on Ω that depends only on coordinates x_s of x for which $x \in c$. The pair $\{V_c, c \in C\}$ is set of potentials of the field, whereas Z is the normalizing constant (called partition function), computed as follows:

$$Z =_{def} \sum_{x \in \Omega} \exp[-\frac{U(x)}{T}]$$

The figure 2.4 show the trend of a unimodal exponential function of the form $\exp[-\frac{|x|}{T}]$. Temperature sharper the shape of the function for small values, whereas to bigger values it smooths the form of the function.

Recall that the local characteristics,

$$p(x_s|x_r, r \neq s) = \frac{\pi(x)}{\sum_{x_s \in \Lambda} \pi(x)} s \in S, x \in \Omega$$

There are two main problems, the former is that the joint distribution of the X_s is not apparent, instead, the latter is that is very difficult to identify the local characteristics, for example when a given set if function $\psi(x_s|x_r, \neq s), s \in S, (x_1, \ldots, x_{s_N}) \in \Omega$, are conditional probabilities for some (necessarily unique) distribution on Ω. To overcome these problem the Hammersley and Clifford theorem is used, that proves the MRF and GRF equivalence.

Theorem 1 (Hammersley and Clifford. MRF/GRF equivalence)**.** *Let n he a neighborhood system. Then X is a MRF with respect to n if and only if $P(x) = Pr(X = x)$ is a Gibbs distribution with respect to n.*

To use the local characteristics is very difficult, this equivalence gives a simple way to employ MRFs specifying potentials. In this way the energy function U can be choice freely. Below the formula is showed, which the local characteristics of $P(x)$ are obtained substituting Gibbs equations.

$$p(x_s|x_r, r \neq s) =$$

$$\frac{\exp[\frac{1}{T}\sum_{c \in C} V_c(x)]}{\sum_{x_s} \exp[\frac{1}{T}\sum_{c \in C} V_c(x)]} =$$

$$\frac{\exp[\frac{1}{T}\sum_{c \ni s} V_c(x)]}{\sum_{x_s} \exp[\frac{1}{T}\sum_{c \ni C} V_c(x)]} = \frac{1}{Z_T}\exp[-\frac{U(x)}{T}]$$

The Ising Model

Ising Model is a statistical model originate in mechanical from physicist Ernst Ising. Its main usage has been in the modeling of magnetic materials, but its used in others many applications.

The general energy form of Ising Model is

$$U(x) = \sum_s x_s G_s(x_s) + \sum_{r \neq s} \beta_{r,s} x_r x_s, r, s \in S \tag{2.1}$$

where $\beta_{r,s}$ denotes pre-defined model parameters which may or may not be site dependent.

To better explain the concept, an example is reported. Consider a lattice of N spins in the presence of a magnetic field H. The spins at each site will take values form ± depending on their alignment with the external magnetic field. However, each spin will interact with its nearest neighbors, thus giving rise to an interaction energy. These two components comprise the energy if the system, defined by

$$E = -H\mu \sum_{s} \sigma_s - J \sum_{r,s} \sigma_r \sigma_s \tag{2.2}$$

where μ is the magnetic moment of an individual spin and J is the coupling constant of the system.

If the two spins are the same the product of the nodes is $+1$, otherwise they are different (anti-aligned) and the product is -1. In Ising's experiment the spins are aligned in a certain direction by the magnetic interaction, but same concept can be used in other scenarios like Image Restoration or, as it will be presented here in Anomaly Detection.

Iterated Conditional Modes

Iterated Conditional Modes (ICM) [32] is an approximation of Simulated Annealing (SA) [33] algorithm. In literature ICM algorithm is more used as compared to SA because this last is computationally expensive, but the ICM needs a good initial configuration to obtain a good convergence toward a nearest energy-valley. The ICM algorithm is a easy way to minimize the energy in a Random Field, the step are explained below.

1. Set a good initial configuration ω^0 and set k=0

2. Compute the energy $U(\eta)$ where $\eta \in N_{\omega^k}$. N is the set of configurations of ω which differs al most in one element from the current configuration ω^k

3. Select a configuration in N_{ω^k} with minimal energy

$$\omega^{k+1} = \arg\max_{\eta \in N_{\omega^k}} U(\eta)$$

4. Repeat the step 2 until convergence is reached.

At the end of minimization the configuration is set. To solve the problem of "good initialization", it was used a measure of similarity between the physical quantities manipulated, it is better explained in Section 3

2.2.2 Bayesian Network

Bayesian networks (BN) [34] are a particular graphical model to permit modeling causes and effects scenarios. These networks permit to represent causes and effects via an intuitive graphical representation. The variables in the Bayesian belief network are represented by nodes. The value assumed by node can be a state, or a set of probability value. Nodes are connected with edges that indicate the causality relationship.

The network models the probabilistic relationships between variables of a system, accounting also the their historical informations, and permit to model scenarios where some informations are partially unavailable or uncertain too.

Bayesian networks are being increasingly used in many scenarios where the artificial intelligence is needed and their results are very convincing. Like the inexact human reasoning, they give a representation of knowledge producing not exact result, but given a certain degree of truth about a topic, for example, if the question is ”it will probably rain tomorrow”, the bayesian network will give a value of possibles output, in this case, the values will be associate to two states, it rain or not.

Bayes' Rule

Bayesian Network are based on an rule discovered by Tomas Bayes. The rule show the conditional probability of a cause that caused an effect, Bayes' rule is expressed in as:

$$P(b|a) = \frac{P(a|b)xP(b)}{p(a)} \tag{2.3}$$

where $P(a)$ is the probability of a, and $P(a|b)$ is the probability of a given that b has occurred.

A conditional probability table is associated to each node, it contains the conditional probabilities which represent the likelihoods based on past informations. In statistical a conditional probability exactly is the probability of a variable X in the state x given parent P1 in state p1, P2 in state p2 e so on. For each parent and possible state which it can assume a new entry in the CPT is created.

The probability that a variable can assume in a determinate state consid-

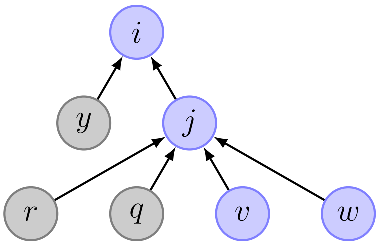

Figure 2.5: Sample Bayesian Network, showing hidden and observed nodes.

ering the actual scenario is defined Belief. In particular a-priori beliefs are a kind of belief calculated only on a prior information, and they are defined only considering the Conditional Probability Tables, whereas the evidence is the information on a current scenario.

Belief Propagation

Belief Propagation (BP) [35, 36, 37] is a well-known algorithm for carrying on inferences on graphical models through *message exchange*; in particular, a message exchanged between two hidden nodes of the model may be interpreted as a piece of information about the state of the former node, as assessed by the latter; such exchange allows to compute the marginal probability of a hidden variable by considering only the observed variables.

In order to provide the theoretical grounds to discuss presented method for modeling sensor nodes behavior, with reference to Figure 2.5, let S indicate the set of nodes of the BN, with $|S| = s$;

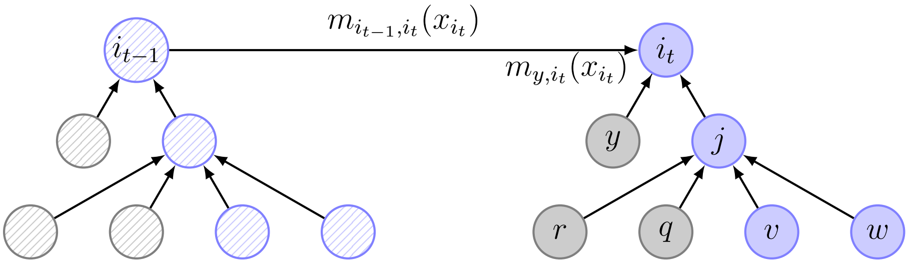

Figure 2.6: A Bayesian network, highlighting message passing between two showing hidden and observed nodes.

Each variable can assume a discrete number of possible states; we will indicate one of the different states of node i as x_i, and similarly for the other nodes. To compute the message between hidden nodes j and i, let H_j and B_j indicate the sets of hidden and observed variables connected to node j; in the depicted case $H_j = \{v, w\}$, and $B_j = \{q, r\}$, with $H_j, B_j \subset S$.

Messages between hidden nodes j and i are of the form:

$$m_{ji}(x_i) \leftarrow \sum_{b \in B_j} \phi_j(x_j, x_b)\psi_{ji}(x_j, x_i) \cdot \prod_{k \in Ngh(j) \backslash i} m_{kj}(x_j) \tag{2.4}$$

where $\phi_j(x_j, x_b)$ and $\psi_{ji}(x_j, x_i)$ represent the *potential functions* between pairs of variables of the graphical model. The former controls the relationship between observed and hidden variables, whereas the latter controls the relationship among hidden variables of the graphical model, and they can be expressed by CPT between node of the net. $Ngh(j)$ represents the set of neighbors of node j.

The "belief" that node i assumes one of its possible values, say x_i, is expressed as follows:

$$b_i(x_i) = \frac{1}{z_i}\phi_i(x_i, y_i) \cdot \prod_{j \in Ngh(j)} m_{ji}(x_i) \tag{2.5}$$

where z_i is a normalization factor. In presented case, we consider that the values of the variables in the model may change over time, so the beliefs are actually re-computed for each instant. Figure 3.8 shows the relationship between two instances of the model at consecutive time instants; a message will also be exchanged between two consecutive instances of node i, and represents an estimate of the state node i will assume at time t, computed at time $t - 1$.

Chapter 3

An Architecture for Anomaly Detection

This chapter describes the proposed system architecture and the mathematic approach to estimate the health status of a wireless sensor node. Following, the architecture of the Anomaly Detection is presented. In particular, the architecture is constituted by two layers, the former implements the physical architecture to acquire environmental data, whereas the latter implements the reasoner to infer the health status of a node. The algorithm for the reasoner consists of two main steps, first it estimates the status of single sensor installed on board of a node, after it estimates the health status of the node. In order to implement the reasoner the mathematical approach is shown afterward. In particular, for the first step, the status of a sensor is evaluated using the MRF technique, in this step the status of a sensor is estimated considering only the physical data acquired by itself. Following the health status of sensor node is evaluated, here the configuration of possibles artificial actuators, which can modify the natural physical quantities, are considered; to infer the value of the health status of the sensor node a Bayesian approach is applied.

3.1 The Logical Organization of the System

This work presents an algorithm for modeling the behavior of sensor nodes, whose main task consists in monitoring typical indoor environmental quantities

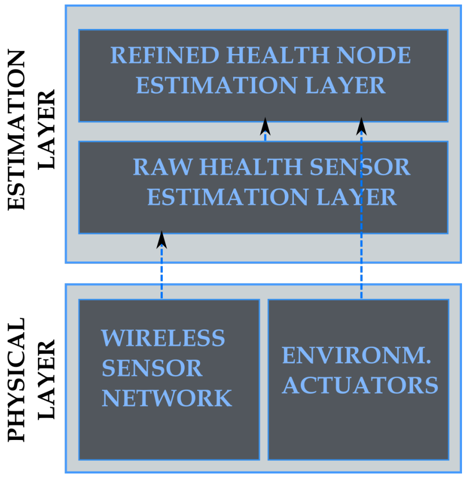

Figure 3.1: Architecture of proposed work, two main layer: PHYSICAL LAYER to acquire environmental information, ESTIMATION LAYER to process the data acquired

in order to detect potential faults; the presented approach will rely exclusively on the analysis of sensed data, with no additional control overhead.

The proposed architecture for anomaly detection is organized in two layers, as showed in Figure 3.1, the former is the PHYSICAL LAYER constituted by a WSN to acquire environmental data and the setting of possible actuators. The latter is the ESTIMATION LAYER, which is formed by two sublayers: the first one is useful to evaluate the raw status of sensors for each considered physical quantity, whereas the second one implements the approach to evaluate the real overall status of node considering also the influence of external factors.

3.1.1 Physical Layer

Figure 3.2 shows the ideal outline for the PHYSICAL LAYER, which consists of a set of hardware technologies to acquire physical environmental quantities and the status of possible actuators. The main technology to acquire physical quantities is assumed to be a WSN. In particular this will be formed by small and smart sensor boards. Sensors installed on board of each node are related to specific environmental quantities such as temperature, humidity, or light.

The status of actuators are acquired by appropriate devices. These are in-

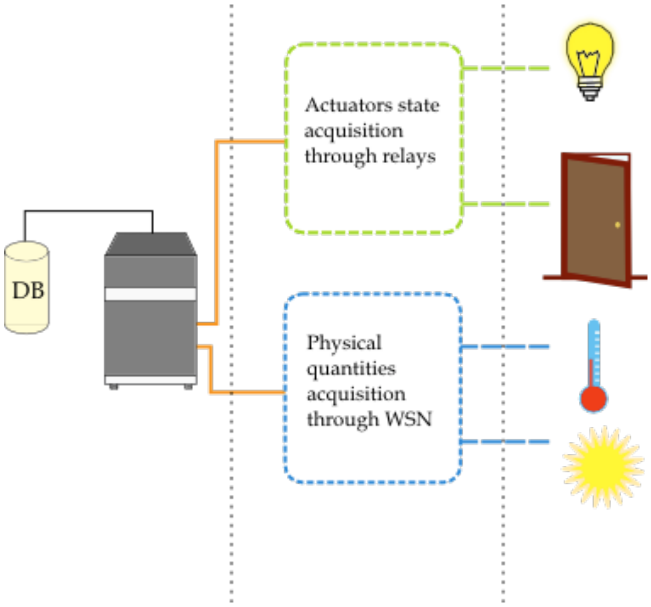
DB
Actuators state acquisition through relays
Physical quantities acquisition through WSN

Figure 3.2: Environmental data and status of actuators are gathered and send to central server.

telligent sensor-to-computer interface modules constituted by a microprocessor, remotely controlled through a simple set of commands. In particular they provide A/D and D/A conversion, and some digital I/O lines for controlling relays and TTL devices. The data acquired by WSN are stored in a database and processed by a software which implements the Estimation Layer as described in the following section.

3.1.2 Estimation Layer

The Estimation Layer is the core of the Anomaly Detection, and artificial intelligence techniques are implemented here. This is constituted by two sublayers, the former estimating the raw health status of a single sensor, while the latter estimating the overall health status of a node. Both layers implement Graphical Model techniques to perform the Anomaly Detection approach.

Raw Sensor Health Estimation Layer

This sublayer reads the data from the database and processes them through a probabilistic approach in order to estimate the raw status of health of each sensor present on a node. The proposed algorithm considers each sensor on board of the node as the generator of a stochastic variable, whose value is to be reliably estimated. Assuming that the considered physical quantities are correlated both in time and in space, the aim is to detect a faulty sensor of a node (i.e. one whose sensors behave as outliers with respect to its neighboring nodes) through an approach based on probabilistic graphical models, namely Markov Random Fields (MRF) [38, 26]. The output of this layer is used by the Refined Health Node Estimation Layer. Details of the algorithm for this layer are presented in Section 3.2.

Refined Node Health Estimation Layer

The Refined Health Node Estimation Layer permits to reliably estimate the health status of a sensor node. It considers the raw status of each sensor, computed via MRF, and the status of the actuators installed in the environment. The actual status of the node is then computed using a probabilistic approach, in particular the WSN node health status is described by a Bayesian Network and computed via Belief Propagation approach. The algorithm is presented in Section 3.3.

3.2 Health sensor estimation

This Section presents the proposed method for assessing the health status of individual sensors [39]; specifically, the intent is to infer the functioning status of a sensor by analyzing its sensed data. A probabilistic approach based on a particular instance of graphical models, namely the Markov Random Fields (MRF) [38, 26] is used in order to classify each sensor according to a binary label representing its status in terms of spatial correlation with respect to sensors for the same physical quantity, on board of nearby nodes. To this aim, groups of nearby sensors are represented as nodes of an undirected graph, as shown in Figure 3.3; starting with an approximation of the health status of each sensor,

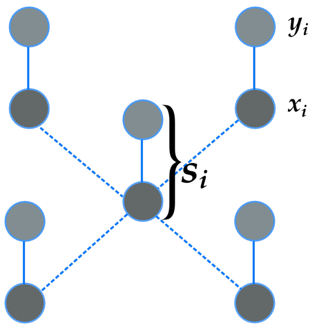

Figure 3.3: Representation of a group of sensors as nodes of a MRF; the health status of each sensor s_i is represented by an observable variable y_i, and a hidden variable x_i.

the proposed method will provide a more reliable estimate for the same status.

3.2.1 Inferring the health status of sensor

The proposed approach is based on the assumption that sensory measurements collected by nearby nodes are similar to each other, due to the intrinsic nature of the considered physical quantities; such similar measurements are then expected to show sufficiently high *spatial* statistical correlation when all sensors are correctly functioning. The proposed method will classify the health status of each sensor obeying this rule as Good, and otherwise it will assume that sensor to be Damaged.

When considering data collected through a WSN, for the sake of simplicity, a single physical environmental quantity will be accounted for.

Let $S = \{s_1, \ldots, s_n\}$ denote the set of sensors located in the considered area, and let us represent the health status of each sensor by means of two stochastic binary variables: Y_i, representing its *observable* health status at a given moment, and X_i, representing its estimated (*hidden*) status. Figure 3.3 shows an example of the configuration for the variables in the MRF.

Intuitively, the former represents the (possibly imprecise) information about a sensor status that can be computed based on the collected measurements, whereas the latter represents the true status, which may not be directly derived from the physical evidence. Now,the aim is to provide a reasonable initialization for all the Y_i; the values for the corresponding X_i will be inferred by minimizing

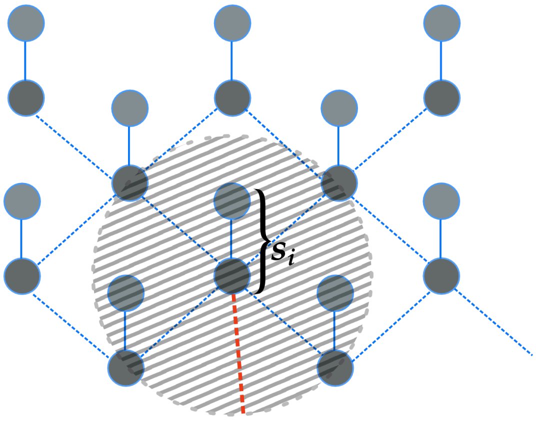

Figure 3.4: Clique is highlighted by gray area, red line indicates the cut-off distance to discovery the subset of nodes.

a globally-defined entropy function.

In order to build the MRF for the considered physical quantity we will work on the undirected graph representing the corresponding sensors; the set of vertices is clearly S, and it will assume that an edge between any two sensors exists if they are "sufficiently close"; the precise definition of closeness is heavily dependent on the chosen scenario; in this case, to define the property of "sufficiently close" it is assumed that nodes have to be close in term of euclidean distance and belong to same environment, for example same room. These principles are to confirm the previous assumption which measurements of close node are spatial-temporal correlated.

Cliques identification

Figure 3.4 shows a clique for the node s_i, it will start by defining a *clique* $C_{s_i} \subseteq S$, which will contain the sensors that most influence the behavior of s_i.

If c_{s_i} indicates the spatial coordinates of sensor s_i, a clique of size ω_{s_i} and composed of sensors distant at most ϑ_{s_i} from s_i will be defined as:

$$\begin{aligned} C_{s_i} = \{s_1, s_2, \ldots, s_{\omega_{s_i}} \quad &: \quad \|c_{s_j} - c_{s_i}\| \leq \vartheta_{s_i} \\ &\wedge \quad \|c_{s_j} - c_{s_k}\| \leq \vartheta_{s_i}, \\ &\quad\;\; \forall j, k = 1, \ldots, s_{\omega_{s_i}}\} \end{aligned}$$

where $\|\cdot\|$ defines the Euclidean distance.

In order for such a clique to exist, we need to determine $\vartheta_{s_i} : C_{s_i} \neq \emptyset$.

In MRFs a clique is useful to simplify the computation of the the probability conditioned of a variable considering the rest of the field. According the Markov properties the conditional probability of a variable can be computed considering only its neighbors, such that the health status of a sensor is $X_i = x_i$ depends only on the sensors within its own clique, as expressed by the following equation:

$$p(x_i \mid \{x_j\}_{s_j \in S - \{s_i\}}) = p(x_i \mid \{x_j\}_{s_j \in C_{s_i}}). \tag{3.1}$$

Estimation of observed variable

Each of the hidden variables will also depend on its respective observable variable; in order to provide a meaningful initialization for each of the latter, it introduces the notion of *temporal* correlation into our method. In detail, it makes the additional assumption that nearby sensors are to show similar trends for their respective measurements across reasonably small time periods.

For each of the sensors in the clique of s_i it will thus consider the window $W(s_j)$ containing the last w readings, where w is dynamically adapted to the considered scenario, as will be explained in Chapter 5. In order to carry out the computations, it is assigned the values -1 and +1 to the labels Damaged and Good respectively, so we will initialize the observable variable for s_i as follows:

$$Y_i = \begin{cases} -1 \quad (\textsc{Damaged}) & \text{if } avgCorr(s_i) \leq 0.3, \\ 1 \quad (\textsc{Good}) & \text{otherwise;} \end{cases} \tag{3.2}$$

where $avgCorr(s_i)$ represents the average correlation between the samples sensed by s_i and those of each of the other sensors in its clique, computed as follows:

$$avgCorr(s_i) = \frac{\sum\limits_{s_j \in C_{s_i}} corr(W(s_i), W(s_j))}{|C_{s_i}|} \tag{3.3}$$

Estimation of hidden variable

It is assuming that a weak correlation (expressed by the 0.3 threshold, according to the Pearson coefficient) denotes an abnormal behavior for the sensor, or in other words, a damaged status.

The estimated health status of a sensor s_i is however represented by its hidden variable X_i; it can make use of the *Hammersley-Clifford* theorem [38] in order to express the probability density function according to the well-known Markov-Gibbs equivalence, as in the following equation:

$$p(x_i, y_i) = \frac{1}{Z} \exp\{\frac{-E(x_i, y_i)}{T_i}\} \tag{3.4}$$

where Z is the *partition function* used for normalization, which may be computed as follows:

$$Z = \sum_i \exp\{\frac{-E(x_i, y_i)}{T_i}\} \tag{3.5}$$

and $E(x_i, y_i)$ is a Hamiltonian function which represents the *energy* of the MRF, following the concept of *Ising Model* [40], and may be computed as follows:

$$E(\mathbf{x}, \mathbf{y}) = -\beta \sum_{i,j} x_i x_j - \eta \sum_i x_i y_i + h \sum_i x_i. \tag{3.6}$$

In Equation 3.6, $\mathbf{x}$ and $\mathbf{y}$ are the sets of all the hidden and observable variables respectively; β and η, are the *coupling parameters* weights between the random variables of the field; namely, the former influences the interaction among "nearby" hidden variables, whereas the latter controls the relationship between each hidden variable and its observable variable; the last parameter, h, weighs the previous status of the hidden variables.

It wants to find out the values for the hidden variables that, given the chosen initial conditions for the observable variables, have the highest probability to minimize the energy function. To this aim we use the algorithm known as Iterated Conditional Mode (ICM) proposed by Besag [32], i.e. a deterministic algorithm which maximizes local conditional probabilities sequentially. It uses the *greedy* strategy in the iterative local maximization to approximate the maximal joint probability of a Markov Random Field. In our case, the ICM sequentially converges to a local maximum of the conditional probability of $p(x_i \,|\, y_i, \{x_j\}_{s_j \in C_{s_i}})$.

It is solved the system of equations using the Lagrange multipliers, after imposing the constraint $\beta^2 + \eta^2 + h^2 = 1$.

Considering the energy

$$E(\mathbf{x}, \mathbf{y}) = -\beta \textstyle\sum_{i,j} x_i x_j - \eta \sum_i x_i y_i + h \sum_i x_i$$

replace the summations with variables A,B and C to make simpler the computation

$$B = \textstyle\sum_{i,j} x_i x_j$$

$$A = \textstyle\sum_i x_i y_i$$

$$C = \textstyle\sum_i x_i$$

now apply the constraint to resolve the equation,

$$\Lambda(\beta, \eta, h, \lambda) = -\beta B - \eta A + hC + \lambda(\beta^2 + \eta^2 + h^2 - 1)$$

at the end partial derivates are computed

$$\frac{\partial \Lambda}{\partial \beta} = -B + 2\lambda\beta \tag{3.7}$$

$$\frac{\partial \Lambda}{\partial \eta} = -A + 2\lambda\eta \tag{3.8}$$

$$\frac{\partial \Lambda}{\partial h} = C + 2\lambda h \tag{3.9}$$

$$\frac{\partial \Lambda}{\partial \lambda} = \beta^2 + \eta^2 + h^2 - 1 \tag{3.10}$$

This approach is used similarly in [41], where an optimization technique to automatically select a set of control parameters for a MRF is presented. Solving the system we obtained the following formulas for the β, η and h.

$$\eta = \sqrt{\frac{1}{(\frac{B}{A})^2 + (\frac{-C}{A})^2 + 1}} \tag{3.11}$$

$$\beta = \frac{B}{A}\eta \tag{3.12}$$

$$h = \frac{-C}{A}\eta \tag{3.13}$$

In proposed method, such parameters are recomputed at each iteration in order to increase adaptability.

Finally, it need to compute the value for the T_i parameter, which represents the *temperature* of the Boltzmann distribution. Since maximum variation for the energy, computed as in Equation 3.6, occurs when a single variable is surrounded by variables of the opposite sign (e.g., a Damaged sensor within a clique of Good sensors) it will use the following formula for T_i:

$$T_i = \frac{var(E_{x_i})}{e^{\theta_i}} \tag{3.14}$$

where $var(E_{x_i})$ is the variance of the energy relative to the latest w readings of sensor s_i, and the parameter θ_i is computed as:

$$\theta_i = \begin{cases} \theta_D & \text{if} \quad x_i = \text{Damaged}, \\ \theta_G & \text{if} \quad x_i = \text{Good}; \end{cases} \tag{3.15}$$

with $\theta_D \ < \theta_G$, so that the Damaged label is preferred in case of higher variance.

Summarizing the algorithm

The algorithm may thus be summarized by the following steps, representing one iteration of the ICM method:

For each sensor s_i:

1. consider the sensor s_i and its clique C_{s_i};
2. compute the value of its observable variable y_i according to Equation 3.2;
3. compute the parameters β, η, h of the energy function;
4. compute the value x_i that currently maximizes Equation 3.4.

At the end of all iterations each sensor is labeled according to the latest x_i.

3.3 Node Health Estimation

In order to assess the operational good standing of a sensor node it is represented its behavior through a Bayesian network [42] (BN) able to model the influence of external factors, so that the overall health status is inferred via belief propagation [43].

The target application domain requires wireless sensor nodes to be equipped with off-the-shelf sensors for measuring common physical quantities in an indoor

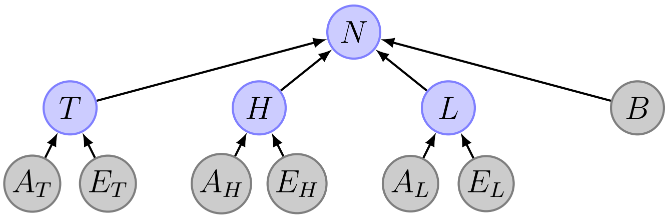

Figure 3.5: The proposed Bayesian network.

environment; moreover, additional common assumptions are made; in particular nodes are supposed to be powered by non renewable energy sources, such as batteries, and the user is allowed to influence the environment where nodes are deployed by operating actuators eventually altering the environmental conditions.

The interest is in modeling the behavior of each sensor node in terms of its ability to provide proper functionalities, by way of a Bayesian network capturing the influence of the surrounding environmental conditions, established by artificial or natural factors, over the sensors on board of the node.

To realize this purpose the Belief Propagation (BP) [35, 36, 37] technique has been used. The BP permits to carry on inferences on graphical models through *message exchange* and it's better explained in the Chapter 2.

3.3.1 Specializing the Model for Indoor Environmental Monitoring

Figure 3.5 shows the structure of the BN used to infer the health status of sensor nodes for environmental monitoring in an indoor environment. Each sensor node is assumed to be equipped with three sensors for measuring light exposure, temperature and relative humidity, respectively; all nodes in the model thus represent binary stochastic variables. In particular, node N in the model represents the health status of one of the deployed sensor nodes; this is the variable to be ultimately inferred. In order to model the behavior of an actual sensor node, proposed model imposes that variable N is influenced by variables L, H, and T which represent the estimators of the operating status of the on-board sensors for light, humidity, and temperature, respectively. Each

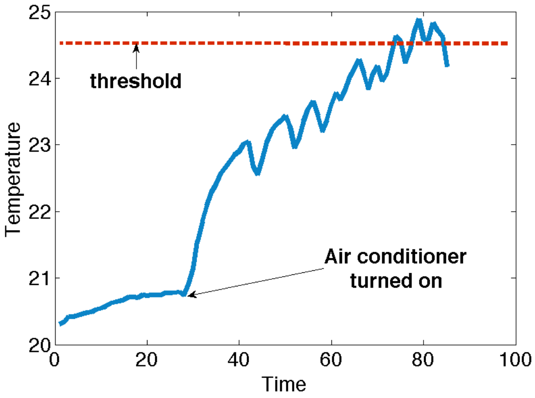

Figure 3.6: Impact of various factors on the functioning of the node: influence of the actuator on temperature

of them models the status of the corresponding sensor also taking into account the operating context, which in this case is represented by the surrounding environmental conditions, as well as the potential influence of actuators over the readings of each sensor.

Variables E_L, E_H, and E_T represent the raw estimators of the health status of the three sensor with respect to their surrounding environment, i.e. their spatial context; meaningful values for such variables are computed via the technique described in section above, where it has been presented a method for assessing the health status of each of the sensors on board of a sensor node by analyzing its readings. Aa probabilistic approach based on Markov Random Fields (MRF) [38] is used, this technique is useful to classify each sensor in terms of the spatial correlation with respect to sensors for the same physical quantity on board of nearby nodes. A healthy sensor will thus be labeled as Good, whereas a faulty sensor will be identified as Damaged.

As is common in indoor working locations, the readings of each environmental sensor is also influenced by human intervention, typically through actuators, such as artificial lighting, or air conditioning systems, which have a direct impact on sensory readings for the corresponding physical quantities. Figure 3.6, for instance, shows the trend of indoor temperature measured by a sensor while the air conditioning system is in function; the sawtooth behavior appears as soon as the temperature approaches the threshold set on the actuator.

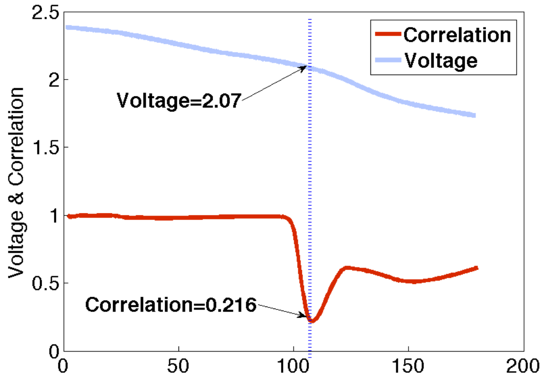

Figure 3.7: Impact of various factors on the functioning of the node: influence of the battery charge.

Variables A_L, A_H, and A_T model the influence of the actuators for light, humidity and temperature respectively. The probabilities associated with such variables are computed with respect to the acquired readings; if the actuator is turned on, the probabilities are computed on the fly by applying Gaussian regression. The μ and σ parameters for the Gaussian are computed based on the latest sensed data sensed by a sensor; subsequent readings are classified with respect to the previously computed Gaussian distribution, thus allowing to estimate $p(x_N = \textsc{Good} \,|\, x_{A_T} = \textsc{On})$. Whenever the actuator is turned off, we assume a uniform distribution for the corresponding variable.

Finally, besides the surrounding environmental conditions, the operating status of a sensor node is also influenced by the charge level of its battery; this is also captured by our model through variable B. Figure 3.7 shows the correlation between the readings of a sensor node with sufficient remaining power, and those of one with low power. The correlation abruptly decreases as soon as the battery voltage approaches a depletion threshold (2.07V in the depicted case). Information about battery depletion is used to model the influence of the battery on the readings in probabilistic terms; in particular, minimum correlation is interpreted as a symptom of low confidence on the inference obtained via the remaining nodes of the BN.

3.3.2 Inferring the Health Status of an Environmental Sensor Node

As previously mentioned, the overall health status of a sensor node is inferred by computing the belief $b_{N(t)}(x_N)$ of the corresponding node N in the BN; in our scenario x_N is a 2-dimensional vector containing the probabilities associated to the two labels, Good and Damaged.

As the model evolves over time, it takes on a configuration depending on the acquired measurements as well as on external perturbing factors. Eventually, the belief about x_N at time t will indicate which of the two possible states is the correct inference for the operating status of the sensor node.

In presented model, variables $E_L, E_H, E_T, A_L, A_H, A_T, B$ are the observed variable, whereas variables N, L, H, T are hidden.

The marginal probability of the hidden variable N_t, in particular is estimated via BP, by applying Equation 2.5, which in the specific case becomes:

$$b_{N(t)}(x_N) = \frac{1}{z_N}\phi_t(x_N, x_B) \cdot \prod_{j \in \{H,L,T,N_{t-1}\}} m_{jN}(x_N) \tag{3.16}$$

where $\phi_t(x_N, x_B)$ is computed via the conditional probability $P(N|B)$, whereas messages are computed by Equation 3.17.

For example for j=T the message is computed as follows:

$$m_{TN_t}(x_{N_t}) \leftarrow \sum_{b \in B_T} \phi_T(x_T, x_b)\psi_{TN_t}(x_T, x_{N_t}) \cdot \prod_{k \in Ngh(T) \setminus N_t} m_{kTj}(x_T) \tag{3.17}$$

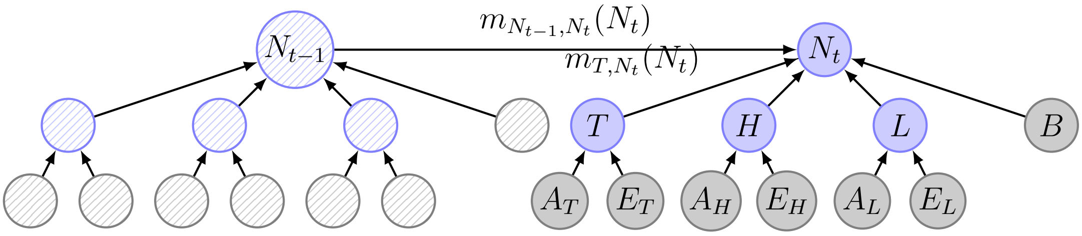

Figure 3.8: A Bayesian network, highlighting message passing between two showing hidden and observed nodes.

where $\phi_T(x_T, x_b)$ and $\psi_{TN_t}(x_T, x_{N_t})$ represent the `potential function` between pairs of variables of the graphical model. The former controls the relationship between observed and hidden variables, whereas the latter controls the relationship among hidden variable of the graphical model, in particular the probability represented by $\psi_{TN_t}(x_T, x_{N_t})$ is computed through the `back-propagation`, in fact after inferring the variable N, the CPT expressed by the ψ is updated, for the first time the CPT are randomly set. $Ngh(T)$ represents the set of neighbors of node T.

It's important to assure the temporal link when the state of the node is computed in order to take in account a previous computed state in next estimation. Figure 3.8 show the evolution of message exchange considering the temporal variable. In fact, to compute the belief of node N at time t requires a message from node N_{t-1}; we assume that such message is null for N_0.

Chapter 4

UML Modeling

This chapter shows the design of the software that implements the proposed work. Before of the programming phase, the presented work has been simulated using Matlab and then it has been developed in Java programming language. The UML standard has been used to model the software. In the following sections UML diagrams are presented. Diagrams shows both static and dynamic behaviors aspect of a software system. Static diagrams describes static structure of the system thought class diagrams, component diagram and deployment diagram. Dynamic behaviors described relationship between system objects and their relative changing of internal states. In this work has been used the state chart diagram presents the dynamic of the approach.

4.1 UML Project

In the last years the Unified Modeling Language (UML) became the standard to model software system. UML permits to software designer to make simpler complex software process improving the quality of system.

The UML standard is very important to design a software system, it is a OMG [44] standard, and it groups concept of software engineering, database and system design. A peculiarity of this modeling language is the independence from all programming languages, and it is usable in heterogeneous applicative domains.

UML diagrams permit to model both structural an behavioral aspect of a software system through nine kinds of diagram. Only diagrams used to model the software presented in this thesis are explained. In particular are presented four kinds of diagrams:

- **Component diagram** is a static diagram of UML standard, it is used to model the high level software component. The component are wired together by using connectors. A connector is useful to specify that the interface of the component is used by another component. In other words, a component uses the service offered by another component which it is connected through specific interface. Component diagram is used to show the internal structure of component, each component can contain one o more classes.

- **Class diagram** is the most popular diagram used in design of object oriented system and it describe the static structure of software. This diagram shows the class of the system and how they collaborate together.

- **Deployment diagram** shows the hardware configuration about the system where the software and middleware are installed. The connections depict which technology is used to wire hardware components, for example RS-232.

- **State chart diagram** represents the functioning flow of the system. This kind of diagram belongs to dynamic diagram and modeling the reactivity of system to specific event.

The diagrams used to design the software relative this thesis are presented below. Four UML diagrams are presented, in particular a component diagram, a deploy diagram, a class diagram and a state chart diagram.

4.1.1 Deployment Diagram

Figure 4.1 shows the deployment diagram of the proposed system. The infrastructure of proposed system is constituted by 4 components. The ADAM and WSN components interact with the environment, whereas the Microserver and Server component process the environmental data.

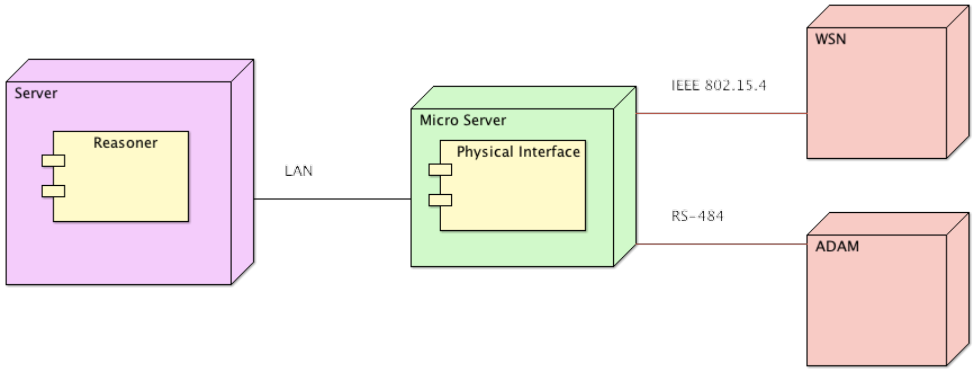

Figure 4.1: Component Diagram for Hardware

ADAM Component

The ADAM device is an intelligent sensor-to-computer interface module containing built-in microprocessor. It can be remotely controlled through a simple set of commands issued in ASCII format and transmitted in RS-485 protocol. It provides signal conditioning, isolation, ranging, A/D and D/A conversion, data comparison, and digital communication functions. In particular it provides digital I/O lines for controlling relays and TTL devices.

WSN Component

The WSN permits to acquire data about physical quantities. This kind of net is used in different scenarios, for example to monitor the condition of an environment, to military purpose, etc. In this work the WSN monitors itself acquiring environmental data, in particular temperature, humidity and light exposure. In the experiment setting the WSN is composed by Mica series and Telos sensor node produced by Crossbow. These are performing sensor, data acquired are sent to microserver via ZigBee.

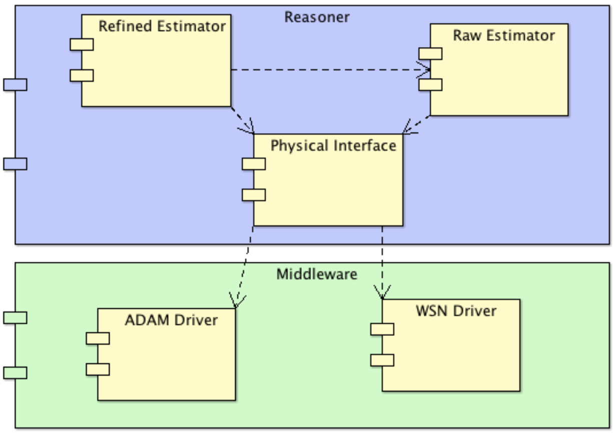

Figure 4.2: Software Component Diagram

Server Component

The Server component is the core of the system. In this component the reasoner and the database are installed. The server receives the environmental data from microserver. The data are stored in a database and processed in order to infer the health status of the Node of the WSN.

Microserver Component

The Microserver is the Stargate NetBridge. This device is an embedded Sensor Network gateway device. Its purpose is to connect Crossbow Sensor Nodes to an existing Ethernet network. It is based on the Intel IXP420 XScale processor running at 266MHz. Stargate NetBridge runs the Debian Linux operating system. This is a full fledged standard Linux distribution for the ARM architecture. This device receives the data from the WSN and sent this to Server.

4.1.2 Software Component Diagram

Figure 4.3 shows the component diagram of the proposed system. The WSN Driver and the ADAM Driver are installed on the Microserver, whereas the remaining components are installed on the server.

Physical Interface

The classes in the Physical Interface component are useful to abstract the physical layer. This component provides environmental data to reasoner component, and it takes care of the managing of problems of the physical communication.

Raw Estimator

This component collects the classes used to estimate the health status of a each sensor installed on node. This module receives environmental data from Physical Interface component and processes them. The result of this component will be refined by the Refined Estimator component.

Refined Estimator

The class grouped in this component are useful to infer the real status of node. This component is interfaced with both the Physical Interface component and Raw Estimator component. The Refined Estimator uses the environmental information about the actuators from Physical Interface, and the health status of sensor from Raw Estimator component.

WSN Driver

This component permits to interface the rest of the system with the WSN. It reads the data from the WSN and makes it available to Physical Interface component.

ADAM Driver

This component permits to interface the rest of the system with the ADAM. It reads the data from the ADAM and makes it available to Physical Interface component.

4.1.3 Class Diagram of System

Figure 4.3 shows the class diagram of proposed system. This diagram shows how the system is composed, it depicts the methods and attributes required by each class.

Actuator Class

This class implements the behavior of the actuator, each actuator has a position, and keeps a state (on/off).

WSN Class

WSN class implements the functionality of a Wireless Sensor Network. It allocates the nodes and actuators that really are dislocated in an environment.

Node Class

Each instance of class Node represents a node belonging to WSN. Each node keeps its health status, the position where it is located and sensors installed on board.

Sensor Class

This class implements the sensor installed on a Node. There is several type of sensors, i.e. temperature, humidity, etc. It records its last sample and its relative timestamp.

RawEstimator Class

This class estimates the health status of a sensor installed on a node. It uses an instance of WSN class and allocates the object for build the structure of Markov Random Field.

RefinedEstimator Class

This class is the core of the system. An RawEstimation object is used to know the health status of each sensors, whereas an instance of WSN is used to know the status of actuators. In this class the algorithm to estimate the health status of a node is implemented.

MarkovRandomField Class

This class implements the inference of algorithm for the Markov random field,explained in Chapter 3. It is used in RawEstimator class.

Clique Class

This class is useful to create the clique. Starting the head of clique the energy is computed. It is allocated in MarkovRandomField class.

BayesianNet Class

BayesianNet class implements the Bayesian network to estimate the health status of a node. It is used in the RefiedEstimator class. For each node an instance of this class is allocated.

4.1.4 State Chart Diagram of System

Figure 4.4 shows the state chart diagram of the proposed system. It explains how the application state changes in order to infer the health status of a node.

Acquiring WSN data State

This is the initial state, the system acquires environmental data, when it has gathered a sufficient number of samples the status is changed.

Sensor health status estimation State

The state machine goes to this state when environmental data are sufficiently gathered. In this state the health status of sensor is evaluate. The MRF technique is implemented in this state.

Sensor state good/damaged

These are transitory states; in order to evaluate the health of the sensor, the state machine goes to one of two possible state which express the good or damaged status of a sensor.

Acquiring status of actuator

After the evaluation of sensor, the state of actuators are acquired. This is a preliminary state before the estimation of the health status of node.

Health status node estimation

In this state the health status of a node is estimate. Here the Bayesian network is implemented.

Node state healthy/damaged

These are transitory states; in order to evaluate of health of the node, the state machine goes to one of two possible health state of a node: healthy or damaged.

Update database

This state is useful to update the new estimated health state to the database. Finally, the end state is reached and the state machine is restarted.

4.2 ER-Diagram

It models concepts or entities and the relationships between them. In database design, this model is used to represent a relational database, where the tables are identified by the entities, whereas the relationship are the keys that point to specific records in related tables.

Figure 4.5 shows the ER diagram for the proposed system, diagram shows five tables, which describe the logical structure of database.

Environment

This table represents the physical environment, i.e. the number of nodes, rooms, ADAM, etc. It is linked with tables NODE and ADAM with relation CONTAINS.

ADAM

This table keeps the ADAM installed in an environment, it describes the ADAM device, each ADAM can have actuators.

RELAY

This table keeps informations about the relay linked to Adam device, and stores the state of the relay.

NODE

It describes the node of a WSN installed in an environment, each node have one or more sensors.

SENSOR

Each entry in this table describes a sensor installed on board of a node. More sensors can be associated to the same Node. For example in this work, nodes have on board sensors of temperature, humidity and light exposure.

READING

Each sensor acquires environmental data. This table stores the data associate to a sensor.

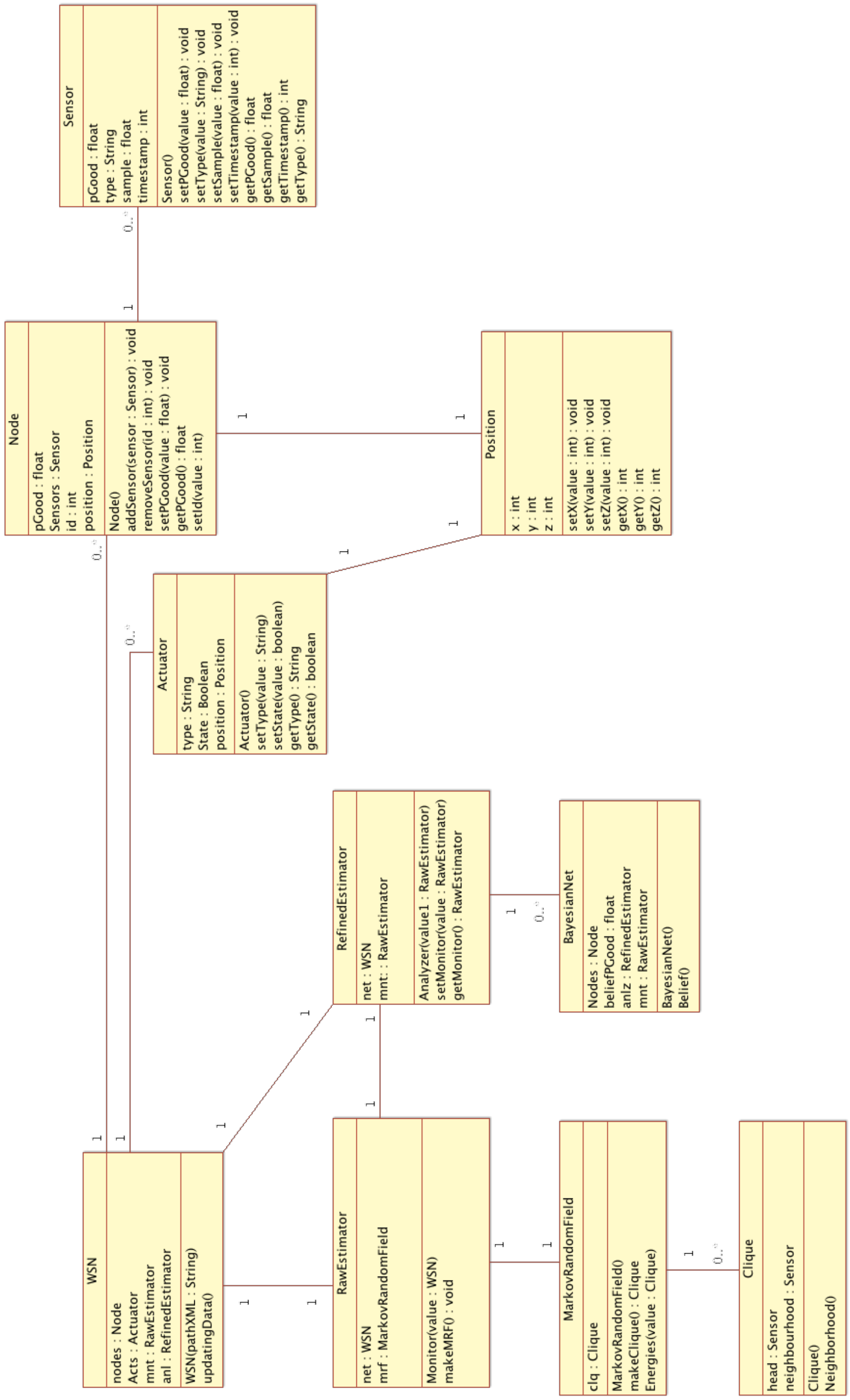

Figure 4.3: Class Diagram

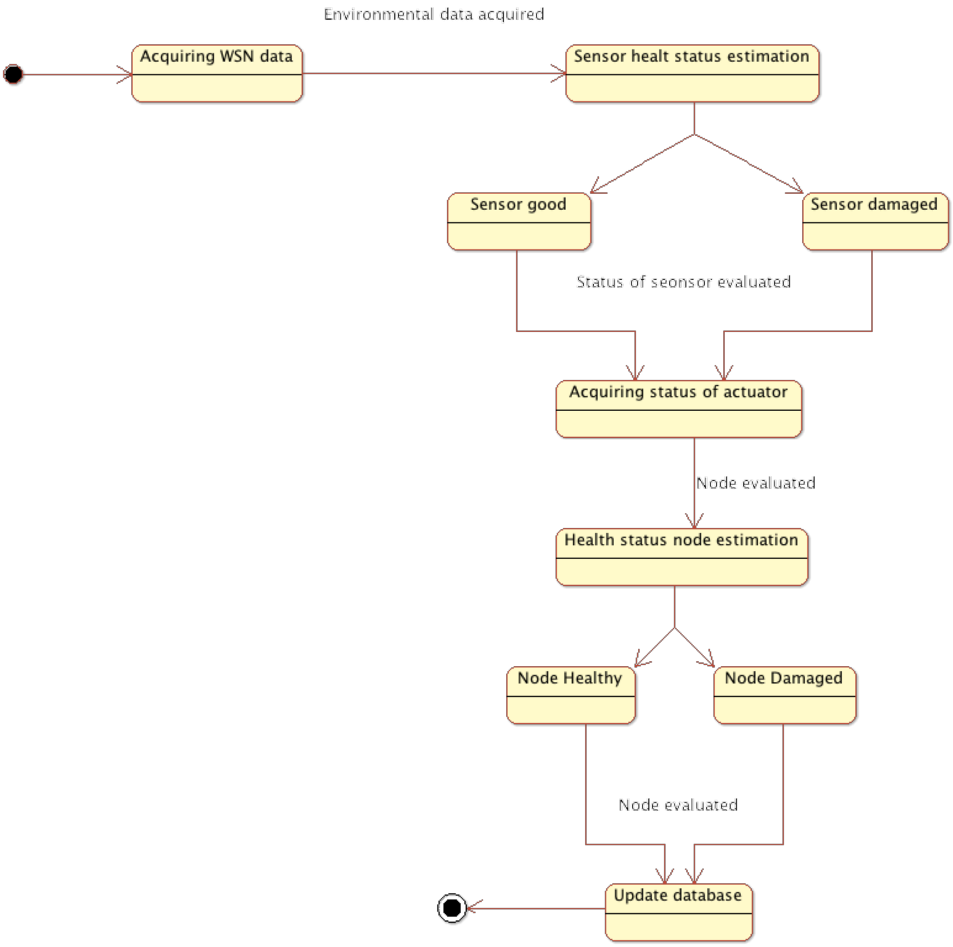
Environmental data acquired
Acquiring WSN data
Sensor healt status estimation
Sensor good
Sensor damaged
Status of seonsor evaluated
Acquiring status of actuator
Node evaluated
Health status node estimation
Node Healthy
Node Damaged
Node evaluated
Update database

Figure 4.4: State Chart Diagram

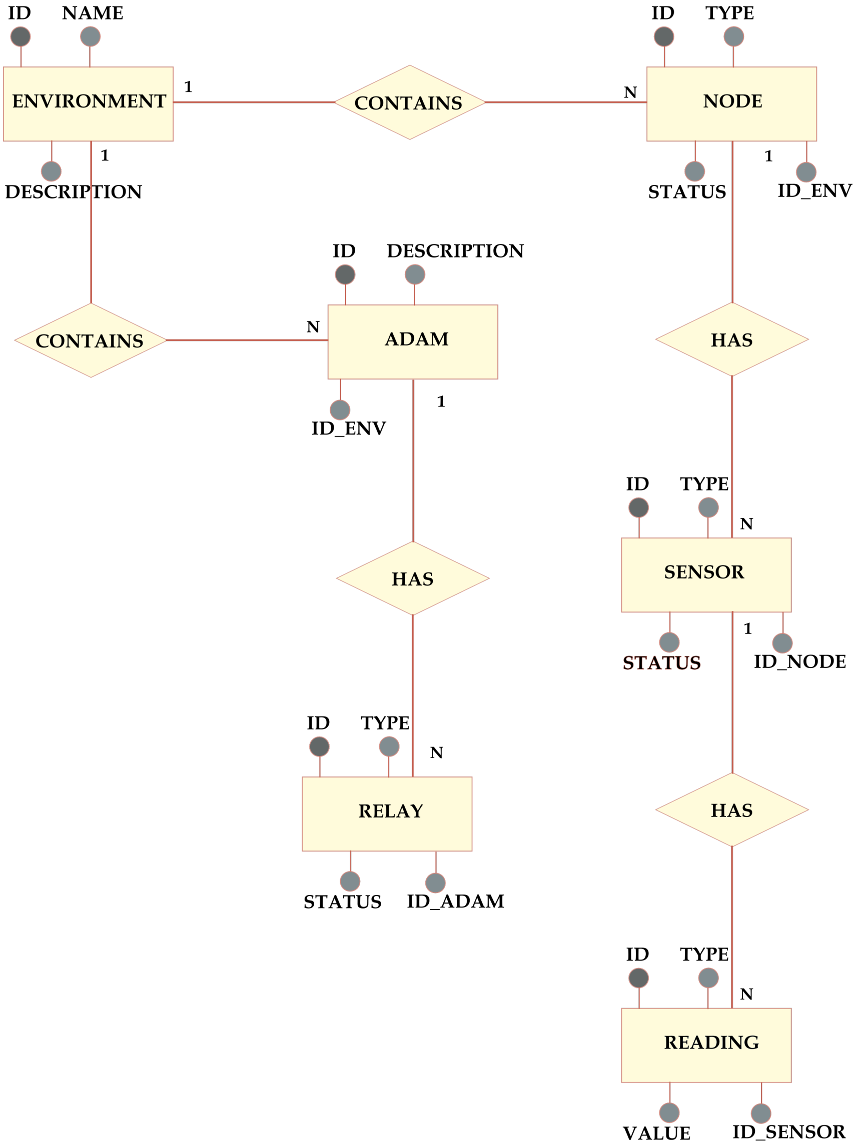

Figure 4.5: ER-Diagram

Chapter 5

Experimental Results

In this chapter results of proposed approach are showed and discussed. In particular the experiments to prove the efficiency of approach are split in two parts. In section 5.1 is shown the experiment for the Raw Estimation Layer, which the Markov Random Field are exploited to infer the status of each sensor installed on node. Three scenarios are presented, each one introduces a different kind of fault that can occur in a sensor.

The experiment for the estimation of the health status of node is explained in section 5.2. This experiment exploits a bayesian network for each node to be evaluated, it considers dataset both influenced by artificial external factors, and simulated error. In this case three scenarios are presented too, the goodness of the algorithm for the first one is computed considering a dataset corrupted by actuator only, in the second one the dataset is influenced by simulated error, whereas in the last one the dataset is influenced by both kinds of factors.

To measure the performance of experiments are used the statistical measure for binary classifiers.

5.1 MRF experiments

In order to assess the validity of our method, it has used a publicly available dataset provided by the Intel Berkeley Research Lab [1], which contains readings collected from 54 sensors between February 28th and April 5th, 2004, via a network of Mica2Dot sensor nodes equipped with weather boards measuring

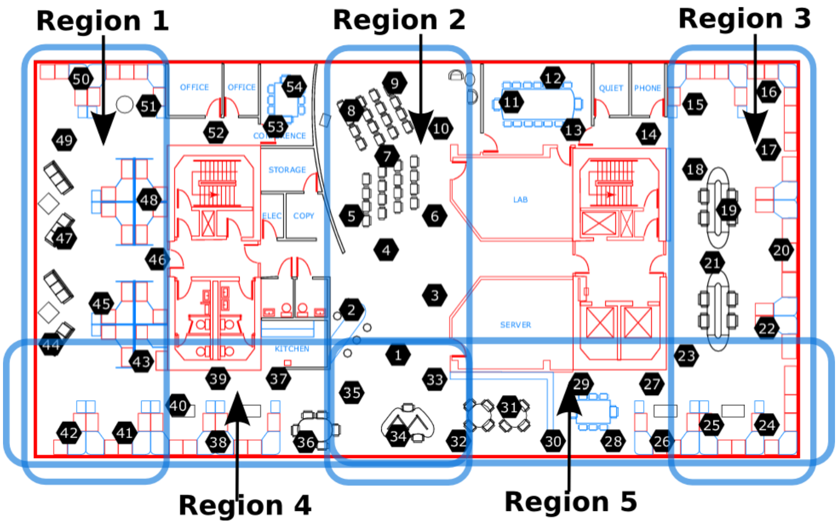

Figure 5.1: Map of the sensor field from Intel Berkeley Research Lab [1], highlighting regions of correlated sensor readings. Red line reads the status of the artificial light, green line reads the status of the air conditioner, whereas the blue line reads the status of window. Sensor nodes capture environmental information and send them to database.

temperature, relative humidity, and ambient light. The authors of [45] report that such data do show significant spatial correlation, and they accordingly divided the nodes into 5 main regions as shown in Figure 5.1.

The algorithm is tested on the readings relative to temperature for a period of 6 days (from March 1st to March 6th, 2004), and considering only 45 sensors, after eliminating those with an insufficient number of readings or falling out of the mentioned regions. In order to build the topology of the MRF for our scenario, it constructed a clique for each sensor, formed at least by four neighbors located within the same regions, as shown in Figure 5.2. Finally, it grouped the considered samples into time slots of 15 minutes, and taking the average value as representative of each slot, in order to disregard the differences in sampling times for various sensors; this resulted in 570 available samples for each sensor.

In order to test the performance of the algorithm, different artificially created faults are superimposed from a subset of the available sensors. A general classification of potential faults is reported in [46]; for the purposes of this section of work, the actual cause of the fault is not relevant, and rather only the

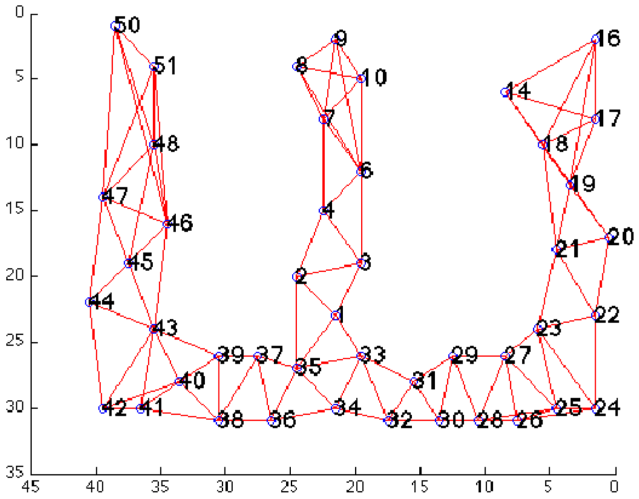

Figure 5.2: Cliques graph for Berkeley Laboratory

actual resulting trend of the considered quantity over time needs to be taken into account, so it has considered here two types of faults (namely, *continuous*, and *discontinuous* faults) obtained by aggregating some of the original classes. *Continuous* faults occur for the entire duration of the experiment; for instance, a sensor can simply produce a constant output, or the sensor readings may happen to be altered by Gaussian noise. *Discontinuous* faults occur at specific time intervals only; we assume that in those intervals the faulty sensors produce a constant output, while returning to normal functioning otherwise. Discontinuous faults are characterized by two parameters: the *duration* of the fault, and the total *number* of its occurrences during the experiment.

This part of approach was assessed by computing two performance metrics for each experiment: *sensitivity* (Se) and *specificity* (Sp). Sensitivity measures the ability of the algorithm, in a particular test, to detect a faulty sensor when it really is, while specificity analogously applies to healthy sensors. They are computed as follows:

$$Se = \frac{Tp}{Tp + Fn}, \tag{5.1}$$

Table 5.1: Continuous Faults

	Constant	**Gaussian $\mathcal{N}(\mu = 0, \sigma = \sigma^*)$**									
σ^*	n/a	0.4	0.8	1.2	1.6	2.0	2.4	2.8	3.2	3.6	4.0
Specificity	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Sensitivity	0.98	0.044	0.41	0.48	0.80	0.85	0.66	0.97	0.93	0.98	0.97

$$Sp = \frac{Tn}{Tn + Fp}; \tag{5.2}$$

where the Tp parameter measures the amount of sensors whose health status is DAMAGED and are actually detected as such (i.e. true positives), the Fp parameter measures the amount of sensors whose health status is GOOD, but are erroneously detected as DAMAGED (i.e. false positives), and analogously for the two remaining parameters.

In order to assess proposed algorithm, 5% of the total available sensors was corrupted by applying one of the previously mentioned faults at a time; "faulty" sensors were chosen randomly, according to a uniform distribution.The experiment was executed each 10 times, and computed the average values for specificity and sensitivity.

Proposed method needs the preliminary setting of two parameters, as explained in Chapter 3: the initial size for the window storing the last w samples for each sensor, and the value indicating the maximum number of allowed iterations for the ICM maximization process. The value of w will be dynamically adapted during each run; namely, it will be linearly increased after each fault detection, and reset to the initial value w_0 when the sensor status becomes GOOD again; in experiments the w_0 parameter is set to 4. The maximum number of iterations was set to 10.

5.1.1 Scenario 1: Dataset influenced by Continuous errors

Figure 5.3 shows the plots representing the trend for temperature measured by healthy sensors, and artificially faulty ones in one run of the algorithm, when

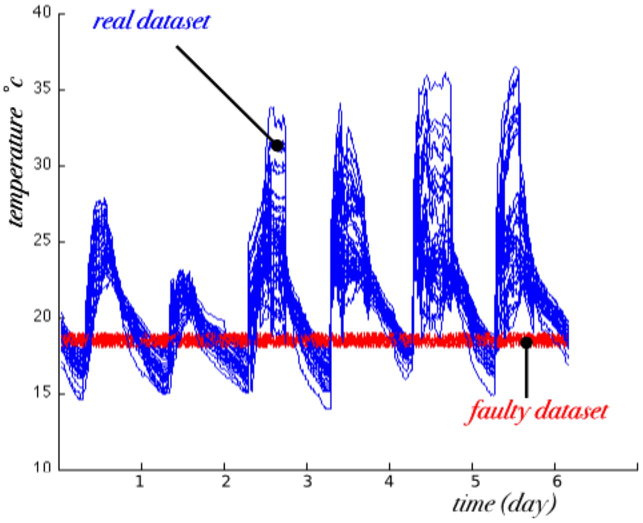

Figure 5.3: Sample of the real dataset, with 5% of the nodes corrupted by *continuous* constant fault.

considering *continuous* faults (i.e. it was assumed that faulty nodes continuously reported a constant value of 18°C corrupted by white noise, with a small variance of 1°C). The three plots reported in Figure 5.6 show the true health status for one of the faulty nodes (constantly DAMAGED in this case), the detected status according to the algorithm, and the corresponding trend for the energy function, as computed according to Equation 3.6. The first column of Table 5.1 shows the resulting performance metrics, which are unsurprisingly good, considering the easily recognizable fault type.

5.1.2 Scenario 2: Dataset influenced by Gaussian errors

The other type of *continuous* fault a *Gaussian* error was considered, it was added to a subset of the sensors; in particular original readings was corrupted with a Gaussian with 0 mean, and increasing variance. The remaining part of Table 5.1 contains the values of the corresponding performance metrics; it shows that *sensitivity* tends to 1 with increasing values for the variance. It is worth noting that significant values for *sensitivity* occur when the variance is greater

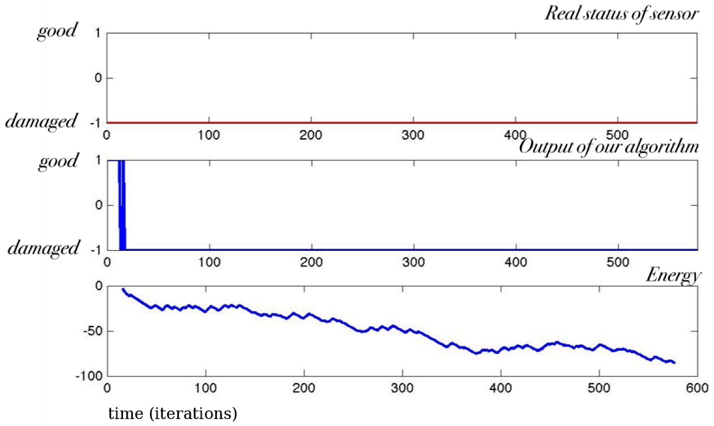

Figure 5.4: Plots showing how the algorithm detects the health status for a faulty sensor.

than 1.6°C, i.e. faulty sensors are correctly detected as soon as the additional error may be distinguished from natural, intrinsic variations of the considered quantity.

5.1.3 Scenario 3: Dataset influenced by discontinuous errors

Faults belonging to the *discontinuous* class cause sensors to produce a constant value of temperature (between 0°C and 5°C in the experiments) during some time intervals, regardless of the natural trend. Figure 5.5 highlights the effect of this type of fault on the dataset.

The algorithm was tested by varying the length of such intervals, and the number of times that a sensor assumes such behavior during the experiment. Table 5.2 reports the performance of the algorithm in two different scenarios: in the former one, we decrease the duration of each fault, while progressively increasing the number of occurrences; we measure the duration of faults as a multiple of the time slots we used, so 96 corresponds to a duration of 1 day (with 6 days corresponding to 570 samples); the latter scenario dually increases

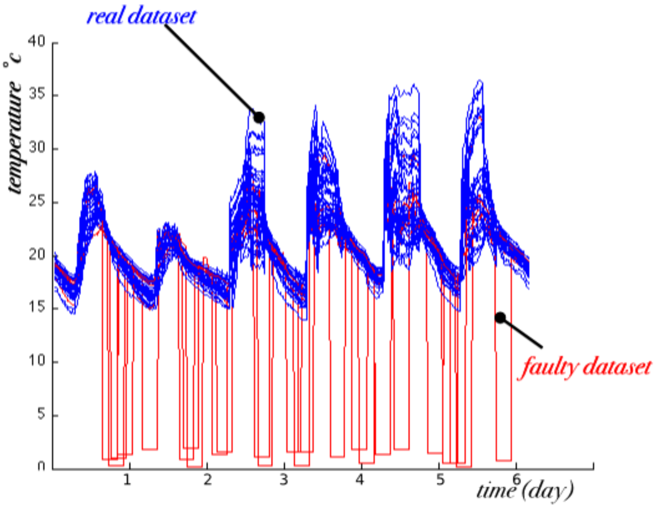

Figure 5.5: Sample of the real dataset, with 5% of the nodes corrupted by *discontinuous* fault.

the duration of the faults, while their occurrences decreases.

Considering the second scenario, it is relevant to highlight that *sensitivity* keeps approaching to 1.0, while the duration of the faults increases; this may be intuitively explained by considering that, as soon as the algorithm identifies a "steadily" faulty sensor, the energy function shows a higher variance than when the sensor behaves correctly, as can be intuitively recognized by considering the parts of the energy plot highlighted by the two dashed rectangles in Figure 5.5.

Also considering Equations 3.4 and 3.14, the parameter θ indirectly influences the final probability $p(x_i, y_i)$; for the experiments, we choose the two possible values for θ so that:

$$\frac{\theta_D}{\theta_G} = 0.1$$

which results in preferring the DAMAGED label.

5.2 BN experiments

In order to assess the performance of the proposed method a simple testbed representing a typical work environment was set up, and was deployed a few

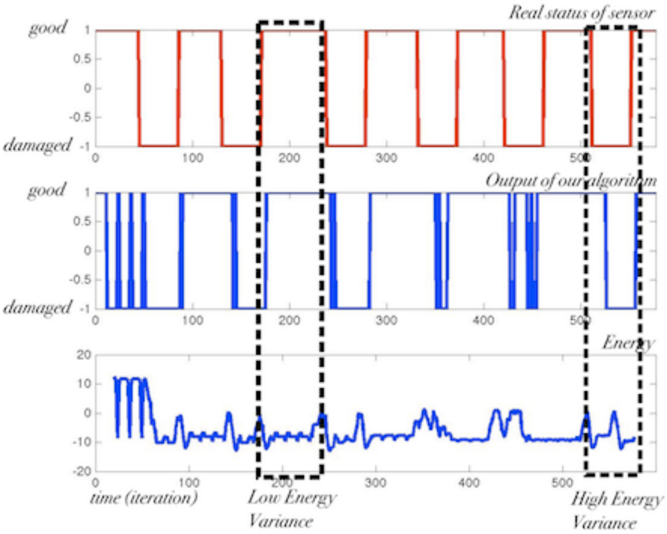

Figure 5.6: Plots showing how the algorithm detects the health status for a faulty sensor.

environmental sensor nodes in an office room, with the purpose of monitoring the common environmental quantities and of detecting potential anomalies. Specifically, 5 sensor nodes are deployed as shown in Figure 5.7. Nodes 1 and 4 have been placed close to the windows, i.e. close to the external source of natural light and heat, whereas nodes 2 and 3 are on top of a bookshelf, in the inner side of the room; node 5 is on the user's desktop, in a central location, also close to the main actuator. The setting also included actuators, namely the air conditioning system (influencing temperature and humidity) and artificial lighting; moreover, sensor readings are influenced by natural factors, e.g. outdoor light coming through the window.

The hardware used for the sensor nodes is commonly available; specifically, the TelosB low-power wireless sensor nodes are used, which are equipped with on-board temperature, humidity and light sensors, as well as an IEEE 802.15.4 compliant transceiver. The main characteristics of the sensors are reported in Table 5.3.

For the experiments, dataset are collected containing measurements acquired during the period ranging from March, 24th to April, 14th 2011. Each sensor

Table 5.2: Discontinuous Faults

	Scenario 1								
Duration	96	48	32	24	19	16	13	12	10
#faults	1	2	3	4	5	6	7	8	9
Specificity	0.98	0.97	0.96	0.95	0.93	0.93	0.92	0.92	0.91
Sensitivity	0.94	0.88	0.83	0.77	0.72	0.69	0.67	0.64	0.61
	Scenario 2								
Duration	1	2	3	4	5	6	7	8	9
#faults	96	48	32	24	19	16	13	12	10
Specificity	0.65	0.68	0.77	0.81	0.85	0.87	0.88	0.89	0.91
Sensitivity	0.28	0.28	0.24	0.36	0.44	0.55	0.57	0.57	0.71

Table 5.3: The sensors used for environmental monitoring, and their characteristics.

Measure	Sensor	Characteristics
Temperature and relative humidity	Sensirion SHT11	Temperature range: -40 °C to +123.8 °C Temp. accuracy: +/- 0.5 °C @ 25 °C Humidity range: 0 to 100% RH Absolute RH accuracy: +/- 3.5% RH Low power consumption (typically 30 μW)
Ambient Light	Taos TSL2550	Range: 400 to 1000 nm Operating range 3.6 to 2.2 volts

node acquired data with a sampling period of 3 minutes; each of the following test scenarios considered an overall time span of 24 hours. Due to its central location, node 5 has been specifically considered as representative for the evaluation of the performance of the proposed algorithm; as will be shown, the influence of all kinds of actuators is more noticeable as compared to the remaining nodes.

Three sample scenarios are considered in order to show the behavior of the proposed method in representative cases; in particular proposed approach was tested by considering a dataset where the influence of actuators on sensor readings was relevant; a dataset artificially corrupted by simulated faults on some of the sensors, and finally a dataset where both actuators and simulated faults were present.

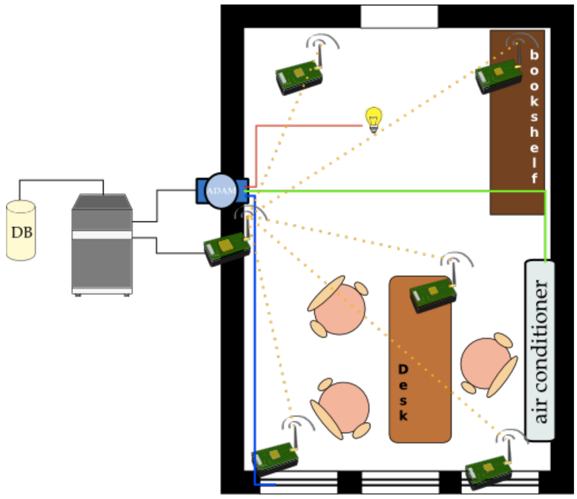

Figure 5.7: A WSN deployed in an office environment, with 5 sensor nodes and one actuator (the air conditioner, AC)

The performance of the proposed approach was quantified by computing two metrics: the *accuracy*, measuring the reliability of the classifier with respect to the detection of Good and Damaged nodes, and the *precision*, which specifically considers the detection of faulty node; they are computed as follows:

$$Ac = \frac{Tn + Tp}{Tp + Fn + Fp + Tn} \tag{5.3}$$

$$Pr = \frac{Tp}{Tp + Fp} \tag{5.4}$$

where Tp measures the amount of nodes whose health status is Good and are actually detected as such (i.e. true positives), Fp measures the amount of nodes whose health status is Good, but are erroneously detected as Damaged (i.e. false positives), and analogously for the two remaining parameters.

5.2.1 Scenario 1: Dataset influenced by actuators

In the first considered scenario, the proposed algorithm processes data influenced solely by the action of the actuators. The Bayesian network correctly identifies data where such influence is relevant, and succeeds in classifying the relative sensors as healthy, even when the underlying MRF-based classi-

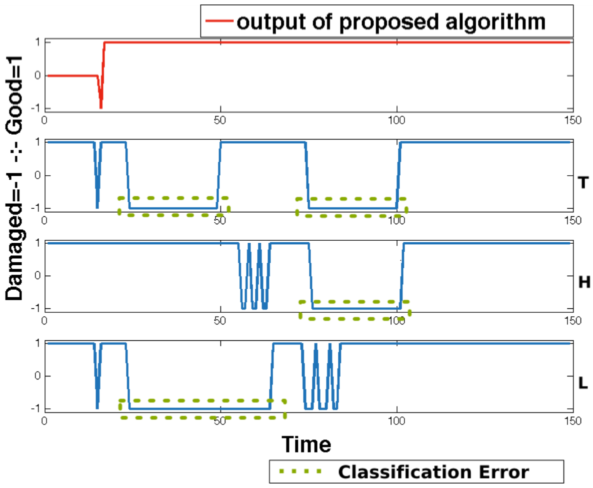

Figure 5.8: Environmental information accounted in the Bayesian classifier, the errors of classification are committed by the classifiers MRF based

fier would trigger an alarm due to the low correlation with "typical" data. Our BN-based classifier provides better performance thanks to the additional information extracted from the environmental context; in particular, in this case the action of the actuators cannot be disregarded.

The outcome of the proposed algorithm is shown in the topmost plot of Figure 5.8; the three other plots in the same Figure show the status of the individual sensors for humidity, temperature, and light as computed by the MRF-based algorithm. The reported plots specifically consider node 5.

Figure 5.8 highlights that the proposed algorithm outperforms the basic MRF-based classifier and is able to correctly classify the node as GOOD even though individual sensors are sometimes reported as DAMAGED (dotted rectangles in the figure).

The performance in terms of accuracy and precision for the proposed Bayesian classifier and for the MRF-based classifiers is reported in the first row of Table 5.4, at the end of this Section.

5.2.2 Scenario 2: Dataset influenced by a simulated faulty

In this scenario the proposed algorithm processes the dataset corrupted by an artificial error only. In particular, a portion of readings has been corrupted

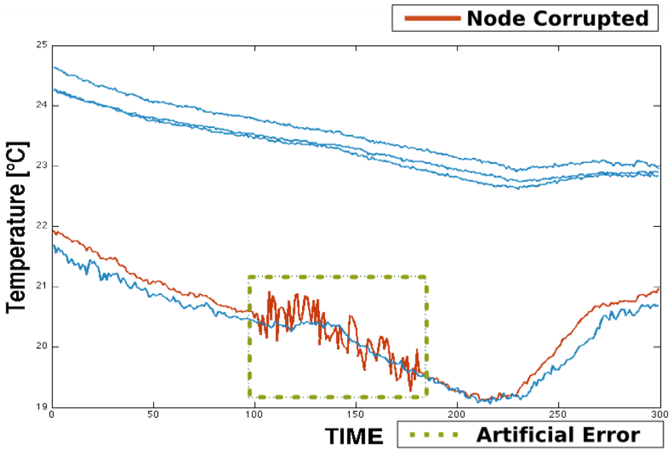

Figure 5.9: Real dataset of temperature perturbed by a Gaussian error

by adding a Gaussian noise with zero mean and variance equal to 5% of the mean of the portion. Figures 5.9 and 5.10 show the original dataset; the dotted rectangle highlights the presence of errors.

In this case the accuracy value for the proposed algorithm is lower than the classifier based on the MRF, due to a transition phase necessary for the algorithm to converge on the exact state.

In the first plot of Figure 5.11, the evolution of the belief about state Good for node 5 is shown. In the others plots, the dotted rectangle surrounds the interval containing the errors for the sensors, which are thus regarded as Damaged. In particular, the interval shown between the leftmost dotted lines indicates the transition phase needed for a node to converge to the Damaged state, when the sensor is corrupted; likewise, a transition phase occurs before returning to state Good, when the last faulty sensor gets back to healthy; this interval is highlighted by the rightmost dotted lines. The transition is due to the fact that the network is time dependent, so that the previous state of a node influences the estimation of next value (through message passing).

Just for this scenario, for MRF-based classifiers, we computed the accuracy and precision for temperature and humidity sensors, since the light sensor is constantly affected by the relative actuator; the performance is shown in the second row of the Table 5.4.

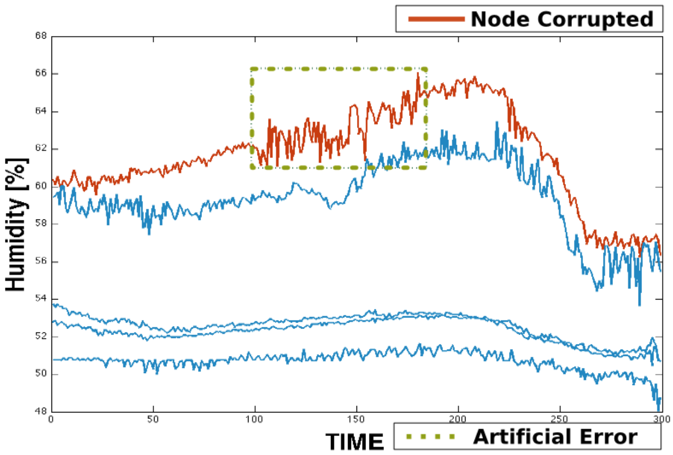

Figure 5.10: Real dataset of humidity perturbed by a Gaussian error

5.2.3 Scenario 3: Dataset influenced by actuators and a simulated error

In this scenario, the dataset used is influenced by the action of the actuators, and by an artificial error. As in the first scenario, the proposed classifier accounts for the environmental information in its reasoning, and correctly identifies the action of the actuators, but similarly to the second scenario, it singles out the artificial error.

The first plot in Figure 5.12 shows the evolution of the belief when the artificial errors occurred on the sensors. This figure rightly shows that the belief of the node decreases only in the proximity of errors, so that the status of the node switching toward DAMAGED value as shown in the first plot of Figure 5.13.

The other plots of Figure 5.13 show that the MRF-based classifier approximately identifies the faulty sensor, signaling the error for a longer time than the Bayesian classifier, which detects the error upon its occurrence. On the third row of Table 5.4, the performance of both kind of classifiers are presented for this scenario.

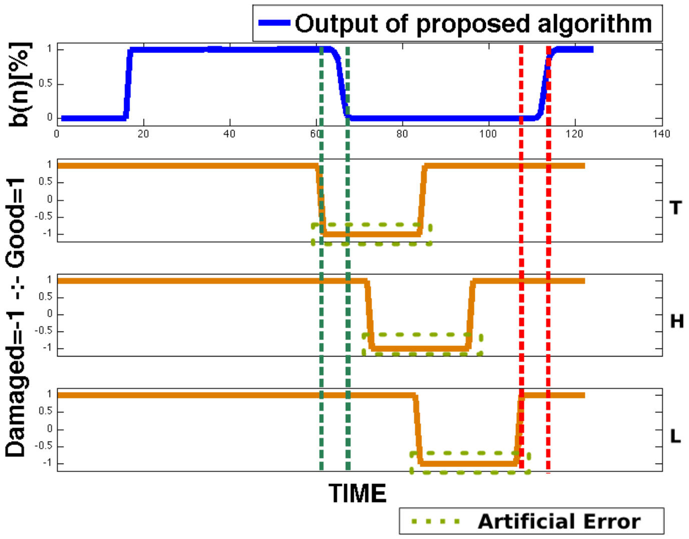

Figure 5.11: Progress of belief of the node 5 during the errors occurred in its sensors, the last three charts indicate the period which the error occur respectively on the sensor of temperature, humidity, and light.

Table 5.4: Performance summary of the experimental scenarios.

	BN Classifier		MRF-based Classifiers					
			T		H		L	
	Ac[%]	Pr[%]	Ac[%]	Pr[%]	Ac[%]	Pr[%]	Ac[%]	Pr[%]
Scenario 1	89	90	77	78	63	64	63	63
Scenario 2	70	93	88	99	78	87	–	–
Scenario 3	78	78	52	51	50	42	80	77

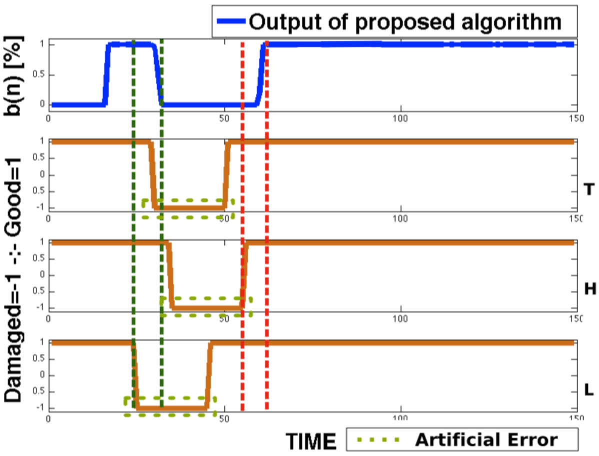

Figure 5.12: Real dataset of humidity perturbed by the air conditioner and by a fault.

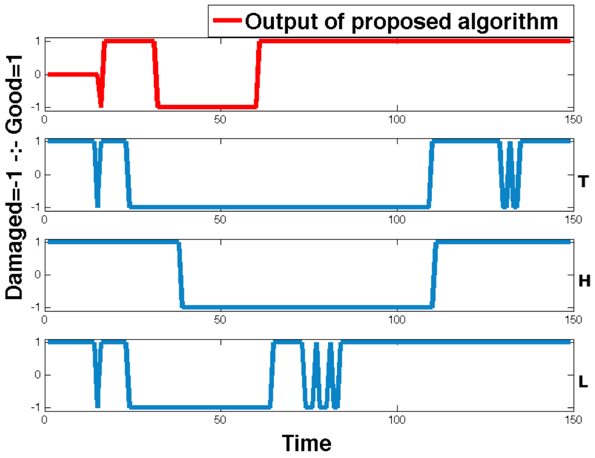

Figure 5.13: Dynamics of the estimate of the status for the classifiers in scenario 3.

Chapter 6

Conclusions

This thesis presented a probabilistic approach to detect anomalies in wireless sensor nodes. In particular the health status of each node belonging to a WSN is estimated. The estimation is computed by considering only the data gathered by the node, while also taking into account those external factors which possibly perturb the natural trend of physical quantities.

This approach can be used in many scenarios, but, in our case, it is developed for an AmI system, showing that the reliability of the whole system is increased.

The proposed approach exploited artificial intelligence techniques, in particular Markov Random Fields and Bayesian Networks, and used smart devices, namely WSN nodes, to acquire data from the environment.

The experiments proved the effectiveness of the proposed approach and appreciable performance is confirmed by the fact that the presented work was tested with two different datasets producing similar results. In particular, the dataset published by the University of Berkeley and data acquired in the our laboratories were used.

The previous chapter showed that this work can be used in different contexts producing similar results using different physical quantities, provided that the acquired quantities conform to the principle of spatio-temporal correlation.

The proposed work is also characterized by its layered structure which makes it extensible and modifiable at any time.

Possible future directions can arise from this work. For instance, starting from the present approach, more precise information may be extracted in order to diagnose the possible causes of the anomaly. Another possible future direction may be the addition of a self-repair component to the presented framework; this new component should actuate precise mechanisms to avoid faults created by an anomaly. Finally, an interesting development would regard the distributed implementation of the proposed method directly within the WSN nodes.

Bibliography

[1] S. Madden, “Intel lab data.” `http://db.csail.mit.edu/labdata/labdata.html`.

[2] E. Aarts and J. L. Encarnação, *True Visions: The Emergence of Ambient Intelligence*. Springer, 2006.

[3] K. Ducatel, M. Bogdanowicz, F. Scapolo, and J.-C. Burgelman, *Scenarios for Ambient Intelligence in 2010, Tech. Rep.* Seville: Information Soc. Technol., Advisory Group (ISTAG), Inst. Prospective Technol. Studies (IPTS), Feb 2001.

[4] A. De Paola, S. Gaglio, G. Lo Re, and M. Ortolani, “Sensor9k: A testbed for designing and experimenting with wsn-based ambient intelligence applications,” *Pervasive and Mobile Computing*, in press.

[5] E. Kandel, J. Schwartz, and T. Jessell, *Essential of Neural Science and Behavior*. New York: Appleton & Lange, 1995.

[6] Santiago Ramon y Cajal, “Structure of the mammalian retina.” Website. `http://en.wikipedia.org/wiki/Retina`.

[7] Wikipedia, “Evolution of the eye.” Website. `http://en.wikipedia.org/wiki/Eye`.

[8] A. De Paola, A. Farruggia, S. Gaglio, G. Lo Re, and M. Ortolani, “Exploiting the human factor in a wsn-based system for ambient intelligence,” in *Complex, Intelligent and Software Intensive Systems, 2009. CISIS’09. International Conference on*, pp. 748–753, IEEE, 2009.

[9] S. Goel, T. Imielinski, and A. Passarella, "Using buddies to live longer in a boring world," in *Proc. IEEE PerCom Workshop*, vol. 422, (Pisa, Italy), pp. 342–346, 2006.

[10] American Medical Association. Website. `http://braininfo.rprc.washington.edu`.

[11] Philips Research. Website. `http://www.research.philips.com/technologies/projects/ami/vision.html`.

[12] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communication Magazine*, vol. 40, pp. 102–114, August 2002.

[13] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. of Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, (Salt Lake City, Utah), May 2001.

[14] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[16] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, pp. 65–72, ACM, 2006.

[17] X.-L. Zhang, F. Zhang, J. Yuan, J. lan Weng, and W. hua Zhang, "Sensor fault diagnosis and location for small and medium-scale wireless sensor networks," in *Natural Computation (ICNC), 2010 Sixth International Conference on*, vol. 7, pp. 3628 –3632, aug. 2010.

[18] R. mao Huang, X. song Qiu, and L. li Ye, "Probability-based fault detection in wireless sensor networks," in *Network and Service Management (CNSM), 2010 International Conference on*, pp. 218 –221, oct. 2010.

[19] B. Krishnamachari and S. Iyengar, “Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks,” *Computers, IEEE Transactions on*, vol. 53, pp. 241 – 250, march 2004.

[20] T. D. Chandra and S. Toueg, “Unreliable failure detectors for reliable distributed systems,” *Journal of the ACM*, vol. 43, pp. 225–267, 1995.

[21] Nicole Sergent and Xavier Défago and Andre Schiper, “Failure Detectors: implementation issues and impact on consensus performance,” tech. rep., École Polytechnique Fédérale de Lausanne, Switzerland, 1999.

[22] A. Farruggia, M. Ortolani, and G. Lo Re, “FDAE: A failure detector for asynchronous events,” in *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*, pp. 197–202, IEEE, 2010.

[23] Oka, Anand and Lampe, Lutz, “Incremental distributed identification of Markov random field models in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2396–2405, 2009.

[24] Yagang Zhang and Jinfang Zhang and Jing Ma and Zengping Wang, “Fault Detection Based on Data Mining Theory,” in *International Workshop on Intelligent Systems and Applications, ISA 2009*, pp. 1–4, may 2009.

[25] T.-Y. Wang, L.-Y. Chang, and P.-Y. Chen, “A collaborative sensor-fault detection scheme for robust distributed estimation in sensor networks,” *IEEE Transactions on Communications*, vol. 57, no. 10, pp. 3045 –3058, 2009.

[26] Dogandzic, A. and Zhang, B., “Distributed Estimation and Detection for Sensor Networks Using Hidden Markov Random Field Models,” *IEEE Transactions on Signal Processing*, vol. 54, pp. 3200–3215, aug. 2006.

[27] R. Brooks, P. Ramanathan, and A. Sayeed, “Distributed target classification and tracking in sensor networks,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163–1171, 2003.

[28] Shekhar, S. and Schrater, P.R. and Vatsavai, R.R. and Weili Wu and Chawla, S., “Spatial contextual classification and prediction models for mining geospatial data,” *IEEE Transactions on Multimedia*, vol. 4, pp. 174 – 188, jun. 2002.

[29] S. Lakshmanan and H. Derin", “Simultaneous parameter estimation and segmentation of gibbs random fields using simulated annealing.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 799–813, 1989.

[30] Stuart Geman and Donald Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 721–741, 1984.

[31] J. Besag, “Spatial Interaction and the Statistical Analysis of Lattice Systems,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 192–236, 1974.

[32] Besag, J., “On the Statistical Analysis of Dirty Pictures,” *Journal of the Royal Statistical Society*, vol. B-48, pp. 259–302, 1986.

[33] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science, Number 4598, 13 May 1983*, vol. 220, 4598, pp. 671–680, 1983.

[34] M. Singh and M. Valtorta, “Construction of bayesian network structures from data: a brief survey and an efficient algorithm,” in *International Journal of Approximate Reasoning*, pp. 259–265, Morgan Kaufmann, 1995.

[35] J. Yedidia, W. Freeman, and Y. Weiss, “Understanding belief propagation and its generalizations,” *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.

[36] A. Brunton and C. Shu, “Belief propagation for panorama generation,” in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pp. 885 –892, june 2006.

[37] V. Savic and S. Zazo, “Nonparametric boxed belief propagation for localization in wireless sensor networks,” in *Sensor Technologies and Applications,*

2009. SENSORCOMM '09. Third International Conference on, pp. 520 –525, june 2009.

[38] Frank, Ove and Strauss, David, "Markov Graphs," *Journal of the American Statistical Association*, vol. 81, no. 395, pp. pp. 832–842, 1986.

[39] A. Farruggia, G. Lo Re, and M. Ortolani, "Detecting faulty wireless sensor nodes through stochastic classification," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pp. 148–153, IEEE, 2011.

[40] R. Kindermann, J. Snell, and A. M. Society, *Markov random fields and their applications*. American Mathematical Society Providence, Rhode Island, 1980.

[41] U. Castellani, A. Fusiello, R. Gherardi, V. Murino, "Automatic selection of MRF control parameters by reactive tabu search," *Image and Vision Computing archive*, vol. 25, November 2007.

[42] A. De Paola, S. Gaglio, G. Lo Re, and M. Ortolani, "Multi-sensor fusion through adaptive bayesian networks," *AI* IA 2011: Artificial Intelligence Around Man and Beyond*, pp. 360–371, 2011.

[43] A. Farruggia, G. Lo Re, and M. Ortolani, "Probabilistic anomaly detection for wireless sensor networks," in *Proceedings of the 12th international conference on Artificial intelligence around man and beyond*, AI*IA'11, pp. 438–444, Springer-Verlag, 2011.

[44] Oriented Management Group. Website. `http://www.omg.org/`.

[45] Guestrin, C. and Bodik, P. and Thibaux, R. and Paskin, M. and Madden, S., "Distributed regression: an efficient framework for modeling sensor network data," *International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004. Third*, pp. 1 – 10, apr. 2004.

[46] S. Zug, A. Dietrich, and J. Kaiser., "An architecture for a dependable distributed sensor system," *IEEE Transactions on Instrumentation and measurement*, 2010.