Investigation of Fault Detection Methods in Wireless Sensor Networks

Charalampos Orfanidis

Kongens Lyngby 2014
IMM-M.Sc-2014

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-M.Sc-2014-

Summary (English)

Wireless sensor networks (WSNs) consist of distributed embedded wireless devices that are used to monitor environmental conditions such as temperature, pressure, sound, etc. Some common characteristics of WSNs are their constrained resources and the fact that they are often deployed in harsh and hostile environments. The sensor nodes of the network can suffer from several faults which could be cause by the environment or from a node malfunction. One consequence of a fault may be the degradation of the communication between the nodes which may affect the whole network topology. For this reason, fault detection is significant for the proper function of a WSN. However, the classical fault detection mechanisms used in regular computer networks cannot be used by WSNs due to constrained resources and extended communication cost, making fault detection a more complex procedure. The resource constrained nature of a WSN calls for an energy-efficient protocol which will achieve the required performance level of the sensor, consuming the least possible amount of energy. A protocol like this can offer extended lifetime to the network and satisfying performance. The faults which may appear are numerous and there are several ways to classify them. There are many scientific articles focusing on fault detection in WSNs, but they do not include the energy-efficiency factor. The intention of this project is to make an analysis of fault detection methods according to their energy-efficiency and overall performance. This analysis is based on existing scientific literature about fault detection techniques on WSNs.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an M.Sc. in Computer Engineering.

The thesis supervisors are Nicola Dragoni, Associate Professor, Departement of Applied Mathematics and Computer Science at DTU, and Yue Zhang, Associate professor, Software Engineering Institute, East China Normal University.

Lyngby, 19-December-2014

Charalampos Orfanidis

Acknowledgements

I would like to thank Nicola Dragoni and Yue Zhang for being my supervisors and guide me through this research. Also, I would like to thank my family for supporting me with every possible manner. Special thanks to Efstratia, for being patient with me during this project and supporting all my crazy ideas. Cheers to Hans-Jacob Sørensen Enemark for the helpful discussions we had in the lab. Special thanks to all of my friends. Finally, very special thanks to Dr. Fafoutis who taught me how to do research and gave the opportunity to work in research for the first time.

"All men by nature desire knowledge"

Aristotle

Contents

Summary (English) **i**

Preface **iii**

Acknowledgements **v**

1 Introduction **3**
1.1 Motivation 4
1.2 Objectives 5
1.3 Structure 5

2 Background **7**
2.1 Wireless Sensor Networks 7
2.1.1 Energy Efficiency in Wireless Sensor Networks 8
2.1.2 Multi-Sink Deployments 9
2.1.3 Topologies 9
2.1.4 Applications 10
2.2 Faults and Fault Detection 12
2.3 Mathematical Background 13
2.3.1 Bayesian Networks 13
2.3.2 Markov Chain 14

3 Related Work **19**

4 Fault Types Classification **21**
4.1 Types of faults in literature 21
4.1.1 Data-Centric viewpoint 21
4.1.2 System-Centric viewpoint 22
4.1.3 Fault-Tolerant Distributed System viewpoint 22

4.1.4 Duration viewpoint 23
4.2 Proposed Fault Type Classification 23
4.2.1 Functional Faults 24
4.2.2 Informational Faults 25
4.2.3 Communicational Faults 26

5 Fault Detection Framework **27**
5.1 Framework Analysis of Fault Detection in Wireless Sensor Networks 28
5.1.1 Information collection 28
5.1.2 Decision making 30
5.2 Approach Description 31
5.2.1 Calculation-Based Approaches 31
5.2.2 Protocol-Based Approaches 33
5.2.3 Hybrid Approaches 34
5.2.4 Remarks 36

6 Evaluation of Fault Detection Approaches **37**
6.1 Evaluation Criteria 37
6.2 Evaluation Data 38
6.3 Discussion 47
6.3.1 Communicational Performance 47
6.3.2 Computational Performance 49

7 Design Guideline **55**
7.1 Communicational Performance Design Guideline 55
7.2 Computational Performance Design Guideline 56

8 Conclusion **57**

Bibliography **59**

List of Figures

2.1 The block diagram of a wireless sensor node 8
2.2 Single-sink and multi-sink WSNs [1] 9
2.3 Single-hop and multi-hop topologies in WSNs [2] 10
2.4 Multi-hop cluster topology in WSNs [2] 10
2.5 Presentation of a fault 12
2.6 Example of a simple Bayesian Network 14
2.7 Example of a Markov Chain 15
2.8 Example of a Hidden Markov Random Field 17

6.1 The mean values of the detection accuracy and false alarm rate over the propose categorization 51
6.2 The mean values of the detection accuracy and false alarm rate regarding the approaches using cluster-based and tree-based topology 51
6.3 The mean values of the detection accuracy and false alarm rate regarding the message exchange patterns 52
6.4 The mean values of the detection accuracy and false alarm rate regarding the calculation methods 53
6.5 The mean values of the detection accuracy and false alarm rate regarding the ASMP-IN-1 criterion 54

List of Tables

2.1 Markov Chain matrix of transition probabilities 16

4.1 Fault Types in WSNs 24

5.1 Design Considerations of Information Collection 29
5.2 Assumptions in Fault Detection Approaches for WSNs 30

6.1 Fault Detection Approaches for WSNs 42
6.2 Notation for the attributes we used from fault-detection approaches 43
6.3 Application-independent criteria for fault detection approaches in WSNs 45
6.4 Application-dependent criteria for fault detection approaches in WSNs 46

Abbreviations

WSN	Wireless Sensor Networks
MD	Message Design
CR	Communication Range
ASMP	Assumption
CM	Calculation Method
OR	Output Range
Ca	Calculation-Based
P	Protocol-Based
Hy	Hybrid
CH	Cluster Head
BS	Base Station
DAG	Directed Acyclic Graph
GM	Graphical Models
JPD	Joint Probability Distribution
MC	Markov Chain
MDP	Markov Decision Process
HMRF	Hidden Markov Random Field
MRF	Markov Random Field
BSN	Body Sensor Network
BN	Bayesian Network
DA	Detection Accuracy
FAR	False Alarm Rate
COMP	Computational Complexity
COMM	Communicational Complexity
FTYPE	Fault Type

Chapter 1

Introduction

The development of low power electronics has led to the progress of the market of the portable electronics. These devices are characterized by their resource constrained nature, in terms of energy, storage and processing. For instance, a system incorporating portable electronics is a *Wireless Sensor Network*.

WSNs are gathering increasing attention as a research topic in academia but also many WSNs-applications in industry are giving solutions to numerous problems (e.g environmental monitoring). The most valuable advantage of WSNs is the fact they can be deployed indoors or outdoors without using a cabling infrastructure and after deployment they do not need maintenance. Of course their lifetime is finite, but they are designed to function for a desired time period without expecting maintenance. A WSN includes a set of wireless sensor nodes and at least one point responsible for collecting the sensed data. The lifetime of a sensor node depends on its battery power. Thus, the network lifetime is finite and depends on the available energy. We have to stress that many times the deployment environment can be harsh, which increases the possibility of a fault occurrence. There is a variety of fault types in WSNs which may affect the performance of the WSN and also consume more power pointlessly.

In this thesis we are doing an analysis on the fault detection methods in WSNs. First we retrieve several scientific papers mentioning a fault detection approach in WSNs, then we propose our fault type classification and see which fault types are included in the selected papers. Next, we propose a framework for fault detection methods and we mention a brief description of the selected approaches. An introduction of an evaluation criteria set and the evaluation of the selected approaches follows. Finally we also provide a design guideline for a fault detection method in WSNs. As far as we know there is not a similar work. The outcome of this thesis may contribute in having a better understanding of fault detection in WSNs and improve the way of designing a fault detection method. We have to mention that in this thesis we do not consider security threats as faults and they are out of the research domain.

1.1 Motivation

As we stressed already, there are a lot of fault types that may occur in a WSN. The consequences can be the decrease of performance or even worse, the complete crash of the WSN. For such error-prone systems fault detection is very significant. Many fault-detection approaches have been suggested in literature for different solutions. Many researches are proposing their own fault type classification for WSNs, thus there are ambiguities on describing a fault from different point of views. A uniform fault type classification would benefit the research in this specific field by providing more coherent solutions.

In fault detection in WSNs every approach is directed too much to the requirements of the application for which it was designed for. An extensive analysis of the fault detection process and proposing a uniform framework for fault detection in WSNs would be beneficial from many points. A newcomer may use this framework analysis as a guide and have a better understanding on the fundamentals of this field.

The applications of WSNs may be used in the context of health monitoring, as in such application the presence of a fault can be crucial, thus the need of evaluating fault detection approaches is important. There is a plethora of fault detection approaches in WSNs directed in different ways, the evaluation of these approaches is important for checking if the specific solution fulfil the asking requirements.

As we mentioned before each WSN application has different requirements and is designed for specific purposes. The existence of a design guideline for fault detection methods would help a potential designer to build a fault detection method according to the desired needs and focus on needed points. For instance, a possible designer may intend to develop a fault detection method for WSNs which does not needs to be extremely accurate but need to have as lower false alarm rate as possible.

1.2 Objectives

During this thesis we deal with an investigation of the fault detection process in WSNs. The objectives of this thesis are described in the following list:

- Analysis of the fault detection process in WSNs
- Evaluation of the selected fault detection approaches in terms of performance and energy-efficiency
- Provision of a design guideline for a fault detection method

1.3 Structure

The structure of this thesis is organized as following: **Chapter 2** gives a background knowledge to the reader in order to be able to follow the next chapters, which are going deep into technical details and analyse the objectives of this thesis. **Chapter 3** deals with the classification of fault types in WSNs. First we present classifications of fault types that we found in literature and then we propose our own classification which is used in this thesis. **Chapter 4** describes in detail the framework of a fault detection process in WSNs divided into phases. The same chapter propose three types of fault detection categories and also presents the fault detection approaches that we selected from the literature. **Chapter 5** lists the evaluation criteria we use for evaluating the selected fault detection approaches, the data we obtained from the approaches organized in tables and a discussion part evaluating the energy efficiency and the performance of the approaches. **Chapter 6** proposes advices for a designer willing to design his own fault detection method for WSNs. **Chapter 7** Concludes the thesis and proposes future directions towards which this work could be improved.

Chapter 2

Background

This chapter provides a fundamental knowledge in WSNs, Fault and Fault Detection and a background section about the mathematical models we encountered more during the research of this thesis. The purpose of this chapter is to familiarize the reader with all the following topics and avoid ambiguities.

2.1 Wireless Sensor Networks

WSNs is one of the of the most appealing topics over the last years in Computer Science in both the academia and the industry. The development of the wireless technologies and microcontrollers made feasible the implementation of a system which is composed from several wireless embedded computing units that are capable of sensing, measuring and store information from the environment they are placed into. Such systems are spatially distributed in a specified area and their objective is to transfer the obtained information to a central unit, called *sink*, for storage and analysis purposes. The wireless communications makes possible the outdoor deployment but usually the outdoor unmonitored environment can be harsh and hostile.

In figure 2.1 it is depicted the architecture of a wireless sensor node. As it is possible to see, it is composed by a microcontroller unit (MCU), a transceiver, a memory module, a sensor, an analog to digital converter (ADC), and the power source, which most of the times is a battery. The wireless sensor node have a set of constraints because of the limited resources. The resource constraints of a sensor node can be described briefly as storage, computing and energy constraints.

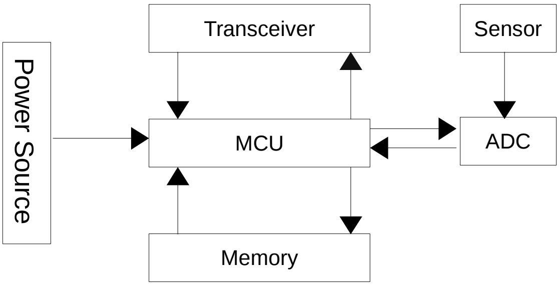

Figure 2.1: The block diagram of a wireless sensor node

2.1.1 Energy Efficiency in Wireless Sensor Networks

The most challenging issue in the area of WSNs is the energy consumption of a sensor node. When a sensor node runs out of energy, it becomes useless. The location of the deployments usually are unreachable, this makes the maintenance of the network more difficult and consequently increase the cost of it. To this end, the primary consideration during the design of WSN or a protocol for WSNs is the efficient management of the available energy, which is also called as *energy-efficiency.*

The radio of the sensor node consumes way more energy than the microcontroller or the other modules of the sensor node. Thus, the energy efficiency is depended to a great degree on the efficient management of the radio [3]. In order to achieve this, the design of an application for WSNs has to be done, considering the regarding constraints. The hardware and the firmware of the sensor nodes, but also the network topology has to be adjusted to the characteristics of the deployed environment and the requirements of the application. Another critical factor for the energy of the WSN is the *duty cycle.* A sensor node has an active and a sleeping state. When the sensor node is in the sleep state, all the components of the node go to the sleep mode. In this way achieves to be active only when it is needed, otherwise it goes to the sleep state to conserve energy. An energy-efficient application in WSNs requires stripping down unnecessary actions which consume extra energy and an analogous duty cycle.

2.1.2 Multi-Sink Deployments

The most important sensor node is the sink, whose objective is to collect all the sensor readings from the sensor nodes for analysis and storage purposes. In addition, we can have a multi-sink deployment which includes multiple sink nodes. The purpose of the sensor nodes are still the same, they have to deliver the sensor readings at one of the sinks. It is assumed that the sink is a common computing device which is connected to the main power supply. Another assumption, is that the sink node has unlimited energy resources and more storage and computing resources than the other sensor nodes.

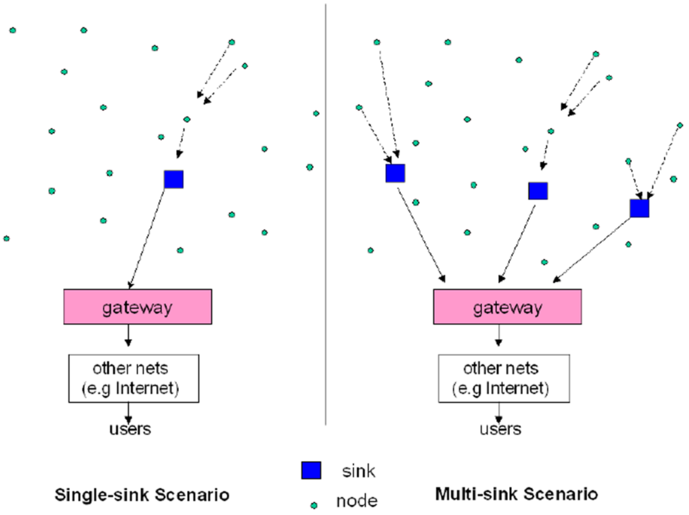

Figure 2.2: Single-sink and multi-sink WSNs [1]

Nevertheless, we can meet other types of nodes in WSNs. For instance we can have Cluster Heads or Leader nodes. These nodes have different functions regarding the requirements of the application. A Cluster Head may be responsible for collecting and forwarding the sensor readings of its cluster or it can be responsible of detecting a fault in the cluster. Sometimes these type of sensor nodes have also increased resources or unlimited energy.

2.1.3 Topologies

In figure 2.3 we can see the two main topologies *single-hop* and *multi-hop*. In the single-hop topology, all the sensor nodes are connected directly to the sink. The duties of the sensor nodes are to transmit and receive only their own packets. On the contrary, in multi-hop topologies the sensor nodes duties are the transmission and reception of their own packets but also forwarding the packets of the sensor nodes which are not connected directly to the sink. A very common multi-hop topology is the cluster-based which is depicted in figure 2.4. In this case the Cluster Heads are responsible for collecting and forwarding the data.

Another case we can see is the *Mobile WSNs*. The network structure in this case is dynamic as both the sink and the sensor nodes can change location during the function of a WSN. The challenges in this specific case are the localization, the navigation but also coverage issues can be increased in such cases.

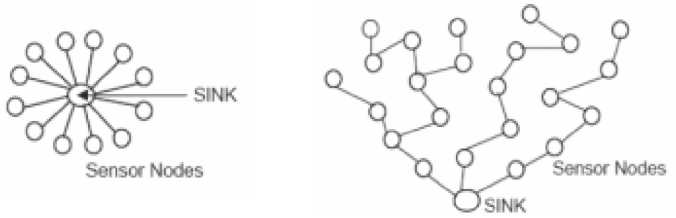

Figure 2.3: Single-hop and multi-hop topologies in WSNs [2]

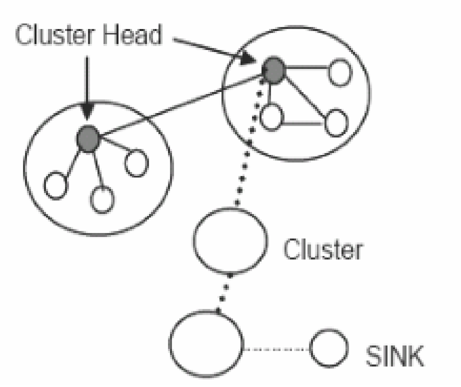

Figure 2.4: Multi-hop cluster topology in WSNs [2]

2.1.4 Applications

The WSNs have been adopted widely in the industry and there are numerous applications today. The sensor reading is different from application to application and is dependent on the sensor unit, it can thermal, biological, mechanical, chemical, optical, etc and depends on what what kind of information is required to extract from the deployed environment. Environmental monitoring is one of the most common applications. For instance, measuring the different environmental statuses like temperature, humidity, light, acidity. Another major field of applications is used for detecting or tracking objects. The possible objects can be humans, animals, vehicles, this field include also other peripherals like cameras, microphones, accelerometers and others.

We can say that the WSNs applications are distinguished in two major categories, monitoring and event detection applications. In monitoring applications we have examples such as environmental monitoring, industrial monitoring, health monitoring and in the event detection we have examples such as detecting or tracking objects, animals, people or vehicles. The event detection applications are used into military industry or public transportation widely. Although we can have applications which include both categories.

The two categories can be discriminated by their communication pattern, the *continuous* and *event driven* [4]. In monitoring applications it is usually used the continuous communication traffic, which is reporting the sensor readings to the sink periodically. In event-detection applications, the communication pattern is not continuous but it is triggered by an event. This particular event may be something not expected, like a human presence in monitored areas for a military application or the discovery of an object from a mobile WSN.

The dominant applications of WSNs are the *environmental monitoring.* It can be indoor or outdoor monitoring. For instance, in U.C. Berkley [5] the indoor environmental conditions such as temperature, light and air pollution are optimized by monitoring them with the sensors of a WSN and keeping them in desired status. In [6], we have an outdoor deployment on Great Duck Island. Here the temperature, the barometric pressure and the humidity were monitored to observe the behaviour of the birds during the change of the climate.

Another application of WSNs is the *animal tracking* which is used in the project mentioned in [7]. Here the objective was to monitor the endangered species of the red wolf. A node was attached in every wolf in order to record their condition and behaviour. The nodes were transmitting their data when a wolf was close to a static sensor.

A very interesting application of WSNs which is getting more and more attention from the researchers is the *WSNs for health monitoring.* In [8], it is described a wearable Wireless Body Area Network for continuous health monitoring. The proposed infrastructure is able to detect abnormal behaviour of a patient and potential knowledge discovery through data mining.

WSNs can be utilized in several military applications such as enemy tracking, battlefield surveillance. The project mentioned in [9], was developed for identifying metallic objects, such as vehicles and armed soldiers and ignoring other objects like civilians.

Development of WSNs has also affect the implementation of Internet of Things [10]. The Internet of things is based on the idea that numerous objects can be uniquely addressed in a way that will allow them to communicate each other in order to reach a common goal.

2.2 Faults and Fault Detection

In [11], it is defined very clearly what is a fault. *A system is said to fail when it cannot meet its promises. In particular, if a distributed system is designed to provide its users with a number of services, the system has failed when one or more of those services cannot be (completely) provided. An error is a part of a system's state that may lead to a failure.* We can see figure 2.5 depicting the above definition. For instance, during the transmission of packets in a network, some packets may arrive with incorrect values(which means a bit having the value 0 instead of 1). Another example is the inability to detect an incoming packet. *The cause of an error is called fault.*

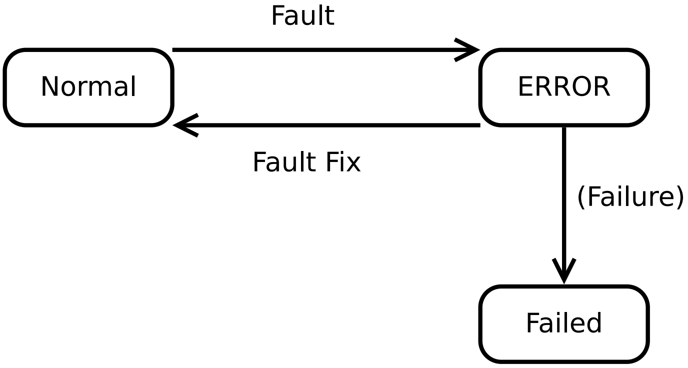

Figure 2.5: Presentation of a fault

The faults in WSNs are specific and have been classified from several researches from different viewpoints. An extended analysis of faults in WSNs is presented in Chapter 3 and it is also proposed a classification which is used in this thesis.

The fault detection mechanism in a distributed system can be defined as following: *The identification of a member in a system which does not deliver the promised services or does not deliver them meeting the timing constraints.* In [11], it is mentioned two main ways to detect a fault in a distributed system: *actively* and *passively.* The former is sending "AreYouAlive" messages to each other and the latter is waiting until a process send you a message informing that it is still alive. In general there are many problems lying in the fault detection. One of them is the attempt to reduce the generation false positives. The false positive usually are generated by using a timeout mechanism in an unreliable network. Another issue is that the aforementioned fault detection mechanisms do not provide enough information about the fault. The previous mechanism will be able to detect only a *crash*, but in the next chapters we will see that there are more faults which can harm our system.

These problems exist in WSNs, but the constrained nature of a WSN introduces more problems and makes the fault detection procedure more challenging.

2.3 Mathematical Background

Several mathematical models are used in fault detection approaches in WSNs. Many of them are probability models which take advantage of the spatial-temporal correlation between the sensors. In this part we present *Bayesian Networks* and *Markov Chain*, two probability models which are used widely in fault detection approaches for WSNs.

2.3.1 Bayesian Networks

Bayesian logic is a field of logic which is applied in decision making and inferential statistics that deals with the probability inference. In other terms, it is using a set of former events to predict future events. According to the probability theory, a rule is defined to clarify an hypothesis by factoring in additional evidence and background information, and resulting a probability which represents the degree that the hypothesis is true.

In [12], it is mentioned that the Bayesian Networks (BNs) can be defined as Graphical Models (GM), used to represent knowledge about an undetermined domain. Every node in the GM stands for a *random variable* and every *edge* for a probabilistic dependence between the regarding random variables. BN can be described better with a GM structure called *directed acyclic graph(DAG).* This interpretation is used for representing and calculating the *Joint Probability Distribution* (JPD).

The structure of DAG includes two sets: the set of nodes and the set of the directed edges. As it was mention before, the nodes represent the random variables and the directed edges the dependencies among the nodes. Thus an edge from node A to node B represents a statistical dependence among the two variables. There are also the terms *parent or ancestor* and *child or descendant* which refer to the fact that if there is a directed edge from node A to node B, then a value taken from node B is depended from a value from node A. In the later case node A is the *parent* or *ancestor* of node B and node B the *child* or *descendant* of node A.

The structure of DAG ensures that there is no node that can be its own ancestor or descendant. A more concrete and formal definition of a BN according to [12] is: a Bayesian network B is an annotated acyclic graph that represents a JPD over a set of random variables $\mathbf{V}$. The network is defined by a pair $\langle G, \Theta \rangle$. G stands for the DAG whose nodes $X_1, X_2, ..., X_n$ represent random variables and edges represent direct dependencies between theses variables. Each node included in G is independent of its non-descendants given its parents on G. Θ represents a set of parameters of the network. Namely the parameter $\theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$ which applies for each realization x_i of X_i conditioned on π_i, the set of parents of X_i in G. Finally JPD is defined uniquely by B over V:

$$P_B(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P_B(X_i|\pi_i) = \prod_{i=1}^{n} \Theta_{X_i|\pi_i}$$

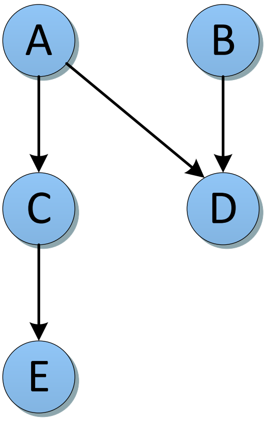

Figure 2.6: Example of a simple Bayesian Network

2.3.2 Markov Chain

According to [13] a Markov Chain(MC) is defined as a set of states $S = \{s_1, s_2, ...s_r\}$ and a process which starts from one state and moves successively. Every move is called *step*, the process can move from s_i to s_j and the probability is denoted as p_{ij}. The probability p_{ij} is independent upon which states the process was before. The probabilities p_{ij} are called *transition probabilities*. An initial probability distribution, between the domain of S, defines the starting state. In general if a Markov chain has r states, then it can be defined by the following equation.

$$p_{ij} = \sum_{k=1}^{r} p_{ik} p_{kj}$$

Figure 2.7 presents an example of a Markov Chain of a student case. The example includes several scenarios and there are also the transition probabilities for moving from state to state. For example a student can start from Class 1 then go to Class 2 and then go to Sleep or Start from Class 1 go to Class 2 then go to Class 3 and then go the Pub.

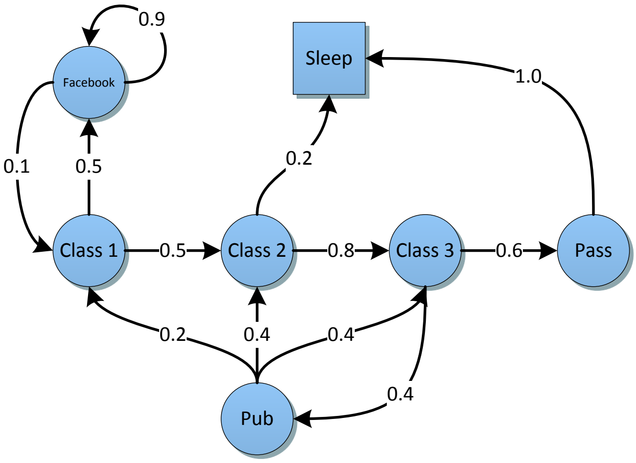

Figure 2.7: Example of a Markov Chain

Markov Decision Process (MDP) and *Markov Random Field* (MRF) are used extensively in Fault Detection techniques in WSNs. MDP is used to model decision making for cases that the result is partially random and partially controlled by the decision maker. MDP is defined as a 4 tuple $(S, A, P(i,j), R(i,j))$. S stands for the set of transition states, A is a finite set of actions. $P_a(i,j) = Pr(i_{t+1} = j | i_t = i, a_t = a)$ is the probability that action a in state s at time t will move to state j at the time $t+1$. $R_a(i,j)$ is the expected reward function which will be received after moving from state i to j. MRF is structured by an undirected graph and contains a set of random variables having Markov

	C1	C2	C3	Pass	Pub	FB	Sleep
C1		0.5				0.5	
C2			0.8				0.2
C3				0.6	0.4		
Pass							1.0
Pub	0.2	0.4	0.4				
FB	0.1					0.9	
Sleep							1

Table 2.1: Markov Chain matrix of transition probabilities

Property. It reminds a Bayesian Network but the difference is that the BN is structured by a directed graph and is acyclic. MRF is represented by an undirected graph and is cyclic.

Markov Chain is another statistical model which is used widely in fault detection in WSNs. Most of the approaches in the literature use as states $S = (s_1, ..., s_n)$ the set of the sensors. The random variables can be the health status of the node, the measured data, several network data or another set of data that we need to use. It depends on which kind of faults is the approach designed to detect. For example, if we want to design an approach for detecting data faults, we have to consider the sensor measurements as a random variable. The next step is to define the number of the neighbors for calculating the probability. Every model use a different number of neighbors and this number is responsible for a balance between the probability accuracy and the energy efficiency. The final step is calculating the probability p according to the model and decide if there is a fault or not regarding to the result.

An approach can also be based on a *Hidden Markov Random Field*(HMRF) model to describe correlations between various attributes of a sensor, such as measured data and real data in order to detect faults on the measured data. The real values can be obtained by the neighbors measurements following the MRF model:

$$\mathbb{P}(X_i|X_{s-i}) = \mathbb{P}(X_i|X_j, j \in N(i))$$

X stands for the real value, $s - i$ stands for the sensors without sensor i and $N(i)$ is the set of neighbors of sensor i. Figure 2.8 illustrates a HMRF model in WSN, the filled nodes represent the real values, the edges between them denote the Markovian dependence and the white nodes represent the measurement values. In other words the edges among X describe the relationships between the neighbors and the edges among the X and Y, the relationships between measured data and real data. According to [14] the probability p can be calculated as follows:

$$\mathbb{P} = (Y_i = X_i + \zeta_i) = \begin{cases} p, & \zeta_i > |\eta| \\ 1 - p, & \zeta_i \leq |\eta| \end{cases}$$

p is the probability of a sensor to be faulty, Y_i is the measured data, ζ_i is the difference from the real value and η is the maximum noise which a sensor value can have.

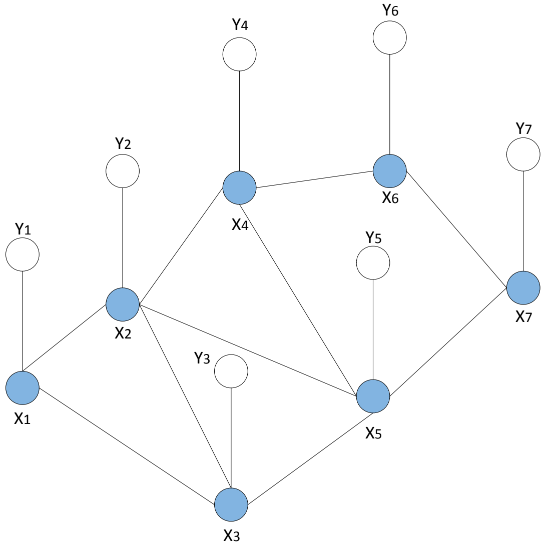

Figure 2.8: Example of a Hidden Markov Random Field

Chapter 3

Related Work

Researchers have given attention to the analysis of fault detection in WSNs but the majority of existing projects on this topic have the form of a survey and only few of them provide an evaluation part.

For instance, in [15] Yu et al. conduct a survey on fault management in WSNs. They divide the fault management process in three phases, *fault detection, diagnosis and recovery.* Regarding the fault detection, they distinguish the fault detection approaches into centralized and distributed and then they classify the distributed ones as following: *node self-detection, neighbour coordination, clustering approach, clustering and distributed detection.* Next they mention an amount of scientific papers selected from the literature which are using the aforementioned fault detection approaches. The next sections analyse the fault diagnosis and recovery phases. Another interesting part is that they propose three different architectures for fault management: centralized, distributed, hierarchical. However, this project does not include any evaluation part.

In [16], Jurdak et al. presents a model including a set of types of WSN anomalies. Next they illustrate a set of anomaly detection strategies for WSNs, which is divided according to the architecture into *centralized, distributed and hybrid.* At last they provide a design guideline for anomaly detection strategies.

Mahapatro et al in [17] adopt a fault type model from [18] and provide their own taxonomy of fault detection techniques which is divided in centralized and distributed approaches. They focus on the distributed approaches and they list them as following:*test-based approaches, neighbor coordination approaches, hier-*

archal approaches, node self-detection approaches, clustering-based-approaches, soft-computing-based approaches, watchdog approaches and probability-based approaches. Afterwards, they provide papers from the scientific literature which are using the aforementioned approaches. Finally, they summarize the characteristics of the selected papers in a table and they conduct a set of comparisons between them.

In [19], the authors model and analyse fault detection and fault tolerance in WSNs. They adopt the fault diagnosis model from [17] and then they implement the algorithms described in [20] and [21] with ns2 simulator, which is a widely-used discrete event simulator for both wired and wireless networks, in order to evaluate their detection accuracy and the false alarm rate. This methodology is comprehensive but in this thesis we are dealing with way more fault detection algorithms and it was impractical to follow such a methodology.

Chapter 4

Fault Types Classification

4.1 Types of faults in literature

It is very common to install a WSN in a hostile environment without having the possibility to maintain it. On such a case, the occurrence of a fault is inevitable, nevertheless the types of faults which may occur are several. Several researchers give a classification of faults according different parameters. For instance, Ni et al. [22] classify some relevant characteristics according to environment, system, and data features. Based on these characteristics they classify faults from *data-centric* and a *system-centric* perspective. Mahapatro et al.[17], propose a classification from the viewpoint of the *the fault-tolerant distributed system* and the *duration*. Before analysing the proposed fault type classification, we give a brief description of the previous fault types given in [22] and [17].

4.1.1 Data-Centric viewpoint

The data-centric viewpoint describe faults that are related to the data readings. More specifically, this viewpoint does not have a description of the underlying cause of each fault and it is easier to define a fault by the characteristics of the sensor reading behaviour. The include fault types are described as following:

- *Outlier*: Isolated data point or sensor unexpectedly distant from models
- *Spike*: Multiple data points with a much greater than expected rate of change
- *Stuck at*: Sensor values experience zero variation for an unexpected length of time
- *Noise*: Sensor values experience unexpectedly high variation or noise

4.1.2 System-Centric viewpoint

The system centric viewpoint, mostly includes faults which are directed to the malfunction of the sensor node. In detail, it describes malfunctions, conditions or faults with a sensor and mention what kind of consequences it will have on the data.

- *Calibration*: Sensor reports values that are offset from the ground truth
- *Connection or Hardware*: A malfunction in the sensor hardware that causes inaccurate data reporting
- *Low battery*: Battery voltage drops to the point where the sensor can no longer confidently report data
- *Environment out of range*: The environment exceeds the sensitivity range of the transducer
- *Clipping*: The sensor maxes out at the limits of the ADC

4.1.3 Fault-Tolerant Distributed System viewpoint

In fault-tolerant distributed system viewpoint we see a different classification based on the behaviour of the failed component.

- *Crash*: A crash faulty sensor node loses its internal state and cannot participate in in-network activities. This is a natural fault [23] that are caused by natural phenomena without human participation.
- *Omission*: A sensor node that does not respond to the sink node on time, fails to send a required message on time, or fails to relay the received message to its neighbour is exhibiting an omission fault

- *Timing*: A timing fault causes the sensor node to respond with the expected value but either too soon, or too late
- *Incorrect Computation*: This refers to the fault that occurs when a sensor node fails to send the true measurement even though the sensing element of the sensor node perceived the true data
- *Fail stop*: The fail-stop fault occurs when a sensor node ceases operation due to depletion of battery and alerts its one-hop neighbours of this fault
- *Authenticated Byzantine*: An authenticated Byzantine fault causes a component to fail in an arbitrary manner that cannot imperceptibly alter an authenticated message
- *Byzantine*: The previous failure classes have specified how a sensor node can be considered to fail in a different domain. It is possible for a sensor node to fail in all the domains in a manner, which is not covered by one of the previous classes. A faulty sensor node in particular may corrupt its local state and send arbitrary messages, including specific messages aimed at bringing down the system. A failed sensor node which produces such an output will be said to be exhibiting an arbitrary failure or Byzantine failure

4.1.4 Duration viewpoint

This duration viewpoint, as it is obvious, classifies the fault types regarding their duration.

- *Permanent*: Permanent faults are software or hardware faults that always produce errors when they are fully exercised [18]
- *Intermittent*: Temporary internal faults
- *Transient*: Temporary external faults

4.2 Proposed Fault Type Classification

In this thesis we classify fault types from a different point of view regarding the components of WSNs. More specifically we focus on three main parts: software and hardware nodes as functional components, sensor readings as informational components and networking parts as communicational components.

Consequently we propose three kind of faults, *functional faults* on malfunctioning sensor nodes, *informational faults* on incorrect sensor readings and *communicational faults* on networking malfunctions. More specifically, in this thesis we define the faults as following:

- *Functional*: Every hardware or software malfunction which prevents the sensor node to deliver the requested services
- *Informational* : Sensor readings that are correctly sent from a sensor node, but deviates from the true value of the monitored phenomenon
- *Communicational* : Every fault which can be caused by the network component of the WSN is considered as communicational fault

The proposed classification is more generous than the aforementioned ones, however we consider that this domain is wider and it can cover more comprehensive the fault type classification in WSNs. Table 4.1 lists the all fault types mentioned above.

The data-centric view is related to the informational faults, the system-centric and fault-tolerant distributed system are related more to the functional faults and the duration is not considered in the proposed classification. The authenticated Byzantine faults and Byzantine faults are security-related issues and are out of scope of this thesis.

Viewpoints	**Fault Types**
Data-Centric	Outlier, Spike, Stuck-at, Noise
System Centric	Calibration, Connection or Hardware, Low Battery Environment out of Range, Clipping
Fault-Tolerant Distributed System	Crash, Omission, Timing, Incorrect Computation Fail-Stop, Authenticated Byzantine, Byzantine
Duration	Transient, Intermittent, Permanent
Components	Functional, Informational, Communicational

Table 4.1: Fault Types in WSNs

4.2.1 Functional Faults

As stated in Section 2.2 a failure happens when something ceases to deliver what is expected of it. In a sensor node this means that some part of it stops operating in a manner that prevents the node from delivering the promised services, which in this case is the production of sensor data and the forwarding of sensor data. In the rest of this subsection we present some examples from papers we picked from the literature.

Jiang [24], propose a fault detection method to detect faults caused by failure of communication or sensing module of the node due to fabrication process problems, extreme environmental conditions, enemy attacks and battery depletion. For the case of [25], it is taken into account as a fault the case when a node has died or it is not able to provide data at all. Another case in [26], fault is defined anything that can be caused by software or hardware problems. In [27], nodes which are not communicating with a notifying message in a defined interval, are considered faulty. Chen et al. [20], consider as fault the complete malfunctioning of a sensor node. Nevertheless, the faulty sensor nodes are still able to communicate process data. Venkataraman et al. [28], propose a fault detection approach in which they consider only faults caused by energy depletion. For the case of [29], faults are defined as anomalies in WSNs, which can be detected as long as you can define your own scenario. In [30], the cause for possible faults is mentioned as sensor outage, which is the case when a sensor node crashes. In [14], [31], [32] fault it is considered the case when a sensor node is crashed. For the case of [33], faults are considered, when sensor nodes are completely damaged or they run out of energy. Finally in [34], Nie et al. classify the possible faults in their approach in energy depletion, sensor fault, radio fault.

4.2.2 Informational Faults

Informational faults can be of different origins, the sensor can be broken, the power supply to the sensor can be out of specification or the environment around the sensor can have changed temporarily. The following paragraphs give examples on cases that we can see these faults.

In [25] fault is defined as a node which is able to communicate but is transmitting inaccurate readings. Khazaei et al. [35] state that sensor nodes which produce incorrect readings are considered faulty. In the case of [14] a fault can be caused by a node which operates properly, but the sensing readings are incorrect. Kamal et al. [36], in order to describe the informational faults they use the classification proposed by Sharma et al. [37] and the considered faults are: constant, short, noise and calibration. Constant fault is described as the case when the sensor node reports a constant faulty value for a long duration. Short is similar with the spike fault, which is mentioned already. Noise and calibration is also mentioned before. In [20], the considered faults are calibration and noise faults. Nguyen et al. [38], propose their own classification which is divided in two classes: discontinuous for faults which occur occasionally and continuous for sensors providing incorrect readings constantly. Continuous includes bias and drift faults. Bias is described as a constant positive or negative offset from the actual value and drift as a positive or negative offset from the actual value but not constant. Discontinuous includes malfunction and random which are similar with outlier and spike accordingly. De [39] develops a fault detection approach to detect faulty sensor readings. In [30], the faults are described as inaccurate readings and the underlying cause is the sensor malfunction. Ni et al. [40], define a fault as a sensor readings which is incorrect regarding a set of sensors which represent the correct trend. Farruggia et al. [41], consider as faulty sensors the ones which provide corrupted data. For the case of [42], as possible faults are considered the short, constant, noise and drift which are mentioned before. In [43] and [44], noise is the considered fault. Lo et al. [45] [46], consider as fault types spike and nonlinearity faults. In a another project by Lo et al. they consider the same fault types, but they consider also the fault types of drift and noise. In [40], Ni et al. evaluate their approach considering outlier and stuck at faults.

4.2.3 Communicational Faults

Communicational Faults may be caused by increased packet drops, high end-to-end delay, coverage issues, broken links or routing failures.

An example is illustrated in [26], which takes into consideration 3 types of faults, ingress drops, routing failures, link failures. The ingress drop is defined as a relationship between the received and transmitted packets of a node. Routing failures and link failures have self explanatory names. The link failure is considered the cause of a fault also in [33]. In [36], faults can be caused by insufficient network coverage to transmit a packet, packet loss or a routing failure. Khazaei et al. [35], state that sensor nodes which fail to communication intermittently are considered faulty. The approach proposed in [14], considers faults as, link failures and route loops. In [34], faults may occur by network congestion or a bad route in the network. Lau et al. in [47], considers as fault when the end-to-end delay of a sensor, exceeds a certain threshold. In [32] link failures and bad routes are considered as faults.

Chapter 5

Fault Detection Framework

According to [15] Fault Management in WSNs is divided into three parts, *Fault Detection, Diagnosis and Recovery. Fault Detection* is the first phase, when an unpredictable failure occurs inside the network and it must be identified properly because there are many types of faults as it was mentioned in the previous chapter. The *Fault Diagnosis* phase includes the identification of the causes, the types and the location of the fault in the network. The final phase, *Fault Recovery,* is the phase on which the identified faults are repaired and cannot affect the network performance any more. We focus on the *Fault Detection* methods in WSNs and after a thorough research in the literature and an extensive analysis, fault detection approaches in WSNs can be distinguished in two classes, *centralized* and *distributed.* The main consideration of this chapter is the distributed fault detection.

In the first section of this chapter we describe the framework of the fault detection procedure in WSNs. First we focus on the first phase *information collection* stressing out the included characteristics. Next, we analyse the next phase of fault detection, *decision making* and in the last section we briefly describe the approaches we found from the picked literature.

5.1 Framework Analysis of Fault Detection in Wireless Sensor Networks

In centralized approaches a *node* with more or unlimited energy and more resources takes the control of the network and is responsible for detecting a fault. The *central node* is responsible for obtaining information from every node, having the role of *information collector* and also the role of the *decision maker*, which means that after collecting the information it is responsible for deciding if a fault occurs. On the other hand, distributed approaches perform the fault detection locally, each node may be a *decision maker* and *information retriever.* In this way less messages are needed with less energy consumption and extended network lifetime. The centralized and distributed approaches have specific differences and similarities. In this section we mention these points and then we analyse the process of a distributed approach by pointing out which factors can play important role in every phase. The following steps describe briefly the process of a distributed and a centralized approach:

- Information collection
- Decision making

The first phase of the Distributed approaches is the *information collection*, that varies from approach to approach. It may be sensor measurements, network metrics or the battery levels. The second phase is the *decision making*, which is the procedure to decide if there is a fault in the network. This decision is taken after processing the obtained data from the previous step.

Centralized approaches require one centralized node which has the roles of *decision maker* and *information collector.* While for distributed approaches, local nodes may take these roles instead of one centralized node. In centralized approaches, the communication range of the communication is always global, however, in distributed approaches, the communication range is local.

5.1.1 Information collection

The communication in WSNs costs a lot of energy but message exchanging is inevitable for detecting a fault in WSNs. Information collection is a procedure which mostly includes message exchanging. In order to have energy efficiency in fault detection, we have to point out first the characteristics of this step that can affect the energy consumption. In this part we emphasize on three aspects: *Message Exchange Pattern* on how to send messages, *Message Design* about what kind of message to send and *Communication Range* on which are the receivers of the messages, which are presented on table 5.1. An explanation of of each part follows.

Characteristics		**Options**	
Message Exchange Pattern		Active Probing	Passive Observing
Message Desing	Content	Status Indication	Sensor Readings
	Size	Binary Bit	User Defined
Communication Range		Global	Local

Table 5.1: Design Considerations of Information Collection

Message Exchange Pattern(MEP) The Message Exchange pattern(MEP) is the way the nodes exchange messages inside the network. Two typical patterns may be used during message exchanging, two-way request-reply and one-way broadcasting. The first one uses pair-wise query-based messages, mostly in hierarchical topologies. In this thesis we call it *active-probing*. The second one is called *passive-observing*, which is more common on flat topologies, with messages sent without requested.

Message Design(MD) MD mainly concerns about the content and the size of the message during the information retrieval step. The content of message may be an environmental measurement such as the temperature, a network metric or a binary variable which indicates the occurrence of an event. The content of message is greatly related to the type of fault that the fault detection approach is looking for. For instance, if we have be a periodic "IAmAlive" message, indicating the health status of the node, most probably the fault detection approach is dealing with functional faults, if the message content is the end-to-end delay of a sensor node, the method is dealing with communicational faults. The size of the message is also an attribute that can affect the performance and the energy efficiency. To this end, it is very important to have a tradeoff between the message size and comprehensive meaning.

Communication Range(CR) The CR can be defined by how many sensors are involved during the information retrieval step. In centralized fault detection most of the times the messages are exchanged among the central node and the nodes in the network. For the case of distributed fault detection approaches the CR may include the one hop neighbours, a set of nodes in a cluster or only one sensor. The CR is critical for distinguishing centralized and fault detection approaches.

5.1.2 Decision making

In order to decide if there is a fault or not, the sensor nodes need an *input* to do the calculation. As it was mentioned in the previous phase of information collection, the input can be obtained from the exchanged messages. The input vary from approach to approach, it may be a sensor reading or a health status. Furthermore, it is related to the context in which the fault detection is running. The context information is always application-depended and it is hard to have comprehensive view. We describe the characteristics of the context information as a list of *assumptions*. The *calculation method* and the *output* of calculation, are the other critical parts of the decision making phase. In the following paragraphs we analyse them respectively:

Assumptions(ASMP) The characteristics of the context of a fault detection approach might have several dimensions. Some of them may be too application-specific to describe. In this part we focus on those which are general enough and organize them according to the components of fault detection in WSN. Except functional, informational and communicational components of WSNs, which were mentioned before, faults themselves are another fundamental component in fault detection. In table 5.2 we illustrate a summary with an indicating name $ASMP\_X\_i$: $ASMP$ stands for the assumption, X stands for the component category, it can be FU for functional components, IN for informational components, CO for communicational components, FA for fault itself and i stands for the number of the assumption.

Label	Description
ASMP-FU-1	The computation for decision making is fault-free or not
ASMP-FU-2	The sensor nodes are mobile or not
ASMP-FU-3	The sensor nodes are heterogeneous or not
ASMP-IN-1	There is a correlation between sensor readings or not
ASMP-IN-2	The sensor readings are fault-free or not
ASMP-CO-1	The communication channels are fault-free or not
ASMP-CO-2	The network has a specific topology or not
ASMP-CO-3	The network needs a certain degree or not
ASMP-FA-1	The duration of fault is considered or not
ASMP-FA-2	The fault is static or not
ASMP-FA-3	There is a correlation between faults or not

Table 5.2: Assumptions in Fault Detection Approaches for WSNs

Calculation Method(CM) Each approach uses a different calculation method for detecting a fault. A fault may be detected by a threshold test, or by complex inferences based on a specific probability model with temporal and spatial correlation considered. On other cases the fault status may be checked by indicating messages, such as "IamAlive" or "Hello" messages.

Output Range(OR) The output range states the fault status of the fault detection method. The content, format and size are always application-specific, but the range of the output is related to the network structure. For example, in flat networks without hierarchy, the output is usually about the node itself. On the contrary, in hierarchical networks, like a tree-based, the fault status may concern the children or the parents of the node.

5.2 Approach Description

A fault detection approach can vary a lot from another. For example, one approach may detect faults by implementing a Bayesian network and another by using a neighbour voting protocol. We propose three categories of fault detection approaches in order to evaluate them accordingly later:

- *Calculation-Based (Ca)*: This category includes fault detection approaches which are based on an algorithm or a mathematical model like Markov Chains or Bayesian Networks
- *Protocol-Based (P)*: These approaches are based on a protocol such as neighbour voting or periodic test with HELLO messages
- *Hybrid (Hy)*: The specific category of approaches may use both a mathematical model and a protocol for detecting a fault or a combination of them.

In the next three subsections, we give a brief description of each fault detection approaches we picked from the literature for this thesis.

5.2.1 Calculation-Based Approaches

De [39] designs a faulty sensor reading detection algorithm based on weighted voting with both distance and reliability used as weight. The reliability here is derived from a localization error detection algorithm with two-way request-reply messages sent between neighbours, i.e. a node sends a hello or dummy message to its neighbours and each neighbour answers a reply message with calculated relative position information included. By this way every node is able to know its position and confidential level. Afterwards a weighted voting algorithm for detecting faulty sensor readings takes place, which exploits the confidence or reliability data from the previous algorithm plus distance. However, how to collect sensor readings for comparison is not mentioned in the paper. Also, this approach has no specific requirement on node degree but it is specific to long-thin topology.

Jiang et al. [24] improves the decision making criteria: for a node and its neighbours which are possibly normal, if the number of test results indicating faulty within this neighbourhood is more than the number of test results indicating normal, then the status of the node is faulty. The improved approach decreases the requirement on the number of neighbours without decreasing the detection accuracy.

Lo et al. [45] [46] use a pair-wise reference-free method based on the ARX model to determine spike and non-linearity fault in sensor readings. Later they continue their work on [48] but after the new features they add the approach belongs to the hybrid category.

Miao et al. [26] deploy a fault detection algorithm in GreenOrbs to detect ingress drops, routing failures, link failures and node failures based on temporal and spatial correlation between system metrics. Temporal detection investigates sudden change in the correlation graph of a node, while spatial detection discovers pattern differences in the graphs of nodes with similarities. Each node in the network periodically send 22 metrics along with sensor readings to the base station. In each time window, correlation graphs are constructed. The longer the time window, the detection accuracy increase with increasing detection delay.

Kim et al. [43] implement a fault detection method for a Body Sensor Network(BSN) called history based method. The former method method works in two steps, first the sensors are divided into multiple motion groups by using the Gaussian Mixture Model Clustering method, and second it is computed the posterior probability of each sensor's input vector and it's nearest cluster set.

In [30], Dereszynski et al. presents a diagnosis approach named Local-Diagnosis(LD2) which is performed by distributed evidence fusion operations. Every node forwards its own tests and the Dempster-Shafer is used for the fusion of the evidences of each node. Finally the LD2 provides the result of the diagnosis.

In [49], a bayesian approach is implemented in order to select a group of non-faulty sensors. Afterwards the sensors data are evaluated with a Neyman-Person test and according the non-faulty set. We have to say that the sensor group is dynamic and can change during the function of the fault detection approach. Also in [40], Ni et al. propose an approach in which all the sensor readings are transmitted to the fusion center which will be evaluated with a Hierarchical Bayesian Space Time(HBST) model in order to define a trusted group of sensors. The next step is evaluate all the received data according the group of trusted sensors.

In [44], it is developed a fault detection method based on a local threshold test. Here it is defined a maximum number of re-observations. Every node when it takes a measurement, makes a threshold test and if it falls in the unreliable range it will make another observation, otherwise it will send it to the fusion center.

Ma et al. [47] proposes a centralized faulty sensor detection technique based on naive Bayes modelling. The nodes are sending their readings and end-to-end time to the sink and the sink detects the status of the network and the faulty sensors.

In [50], it is illustrated a fault detection method which is based on an intelligent stationary agent named ATLAS. The nodes are sending their data to the agent by using a reverse multicast tree topology and the agent detects faults by using the Expectation Maximization algorithm.

In [34], Nie et al. present a fault detection framework which focus on the de-congestion of the network. All the sensed data are directed to the Base Station(BS), where they are checked by using a self-learning failure knowledge library. Finally, the root causes of the faults are identified and the the detection procedure is complete.

In [51], it is presented a fault detection approach for medical WSNs. The proposed fault detection method adopts the decision tree algorithm in order to detect a fault into the network. The specific method is based on the fact that the physiological results are correlated and strange changes are able to be detected and enable the alarm for a possible fault.

5.2.2 Protocol-Based Approaches

Venkataraman et al. [28] deal with failures due to energy exhaustion to keep connections in a cluster. They define two kinds of messages for every node in a cluster to its parent and children nodes: a hello message including location, energy, and node ID for indicating the existence of a node, and a fail report message, sent by a node whose energy is going to be exhausted, triggering the failure recovery process. The detection of energy exhaustion is done by simply checking the current energy level.

Taleb et al. [25], adopt the De Bruijn graph in constructing multi-layer clusters. A cluster header detects faulty leaf nodes by sending test packets within the cluster and comparing test results with expected values.

In [27], the nodes are reporting their sensed data during their timeslot to the Cluster Head(CH), if they do not sense any data they have to notify the

CH that they are still alive with a small special packet. If the CH does not receive neither data nor the special packet assumes that the specific node is either crashed or out of its range(because the nodes are mobile). Then the CH notifies the Base Station(BS) by sending it the ID of the node.

In [33], a message mechanism between clusters undertakes the detection of faulty nodes. In detail, every CH sends a message to its neighbour CHs to check their status in a defined frequency. If the CHs do not answer, they are treated as faulty and then it is sent another message from another CH to check if the fault is a broken link or complete crash.

In [29], Ramassany et al. proposes a fault detection method based on a defined scenario(automatic generated or defined by the user). The scenario is executed in the WSN and and a set of observers nodes are checking the outputs of the nodes and detect the faulty nodes according the time constraints. Liu et al. in [32] develop a fault detection approach based on Finite State Machines(FSM). The current state is defined by the historical states of the system. The FSM model is generalized and local evidence is used on each sensor nodes as inputs for the scope of fault detection. In [52] a fault detection approach is described, in which every sensor identifies its local fault status based on comparisons of sensed readings of its neighbours. Threshold tests and aggregation of the decision, takes place later to complete the fault detection.

5.2.3 Hybrid Approaches

In [48], Lo et al. divide sensor nodes into arbitrary groups and detect informational faults including spike, non- linear transduction, mean drift, and excessive noise, based on group testing and Kalman filtering. It is unclear how the messages are exchanged between pair-wise nodes or arbitrary group members. Both approaches have no requirement on topology and node degree.

Chen et al. [20] first check the differences of sensor readings between a sensor node and its neighbours. Then the sensor node make decision on its status to be faulty or fault-free by calculating its tendency value Ti with several times of adjustment. Each sensor node use broadcasting to send sensor readings and tendency values to its neighbours. This approach does not have any constraints on topology and it has a high detection accuracy with the requirements on the number of neighbours and higher communication overhead due to several rounds of message exchanging.

For the case of [35], the network is partitioned into clusters and the faults are discovered by comparisons among the neighbours and a CH, which is defined by some threshold tests.

Fang et al. [42] design a two-tiered data validation framework with a two-phase in-network, hierarchical, Demand-based, Adaptive Fault Detection(DAFD) method. During the learning phase, tier-one (local) model and tier-two (spatial) model are established in each node and between local nodes. The operational phase use the above two models to check sensor readings, to determine faulty data and use feedback of the spatial model part to update local model. They also design an adaptive spatial validation selection mechanism to use either group voting or singular spatial validation for detecting faulty data. The number of message exchanges is based on the size of verifier set, data collection window, spatial verification demand, and spatial model update frequency. The approach demonstrates good detection accuracy during the evaluation with consideration of faults like: short, constant, noise, and drift. They use group voting between neighbours to detect informational faults such as short, constant, fault detection is achieved by a process called majority voting.

In [53], it is presented an approach for detecting arbitrary types of faults, based on a feature called mutual divergence. Each node decides locally if it is faulty or not based on which mutual divergence is lower according to the neighbourhood. Another proposed approach in this paper can offer higher accuracy. This approach works as following: every node sends its uniform distribution to its neighbours, in this way a sensor obtain N samples from the distribution to update his own distribution and the calculate the probability to be faulty. If the probability is higher than a certain threshold then the sensor node is faulty.

In [38] Nguyen et al. propose an approach in which a group voting technique and a time series data analysis cooperate, in order ensure the accuracy and reliability of sensor data in large scale deployments.

In [31], it is used a Naive Bayesian Model which encodes the probabilistic correlation between a set of state attributes and root causes. A fusion tree is also used to determine if a node has crashed or not. For example, if some local state values presents strange behaviour a diagnosis process is responsible to check that whether this neighbour has crashed or not.

The research by Gao et al. [14] introduces a fault detection method in WSNs which is based on Hidden Markov Random Field model. Each node uses this model and its neighbour readings in order to detect if it is faulty or not. Furthermore a weighted confidence technique is used to ensure higher accuracy to the results of the model.

In [41], it is developed a fault detection by exploiting the assumption that the sensor readings spatial correlated. Farrugia et. al develop a method which uses a Markov Random Field(MRF) in combination with neighbours data and according the degree of the average correlation, characterize if a sensor node is faulty or not.

Snoussi et al. [54] propose a method for online change detection in WSNs. The method is implemented as following: leader nodes are selected which are responsible to update the system discrete state. The leader node is exchanging statistics with its collaborating nodes based on the fact that there is a spatial correlation between them. We have to mention that the selection of the leader node is temporal and is based on the trade-off between information data relevance and compression loss under the communication constraints.

Warriach et al [55] illustrate a fault detection method which is a combination of 3 other methods, namely rule-based, learning-based and estimation-based methods. The rule-based exploit domain and expert knowledge in order to construct heuristic rules for detecting the faults. Estimation methods use the spatial-temporal correlations between to predict the normal behaviour of a sensor and then identify a faulty measurement. Finally, learning-based methods use a hidden Markov model and by using the measured data calculate statically if a reading is faulty or not.

Kamal et al. [36] developed a fault detection method for ensuring the reliability in a WSN. More specifically they introduced a mechanism in which the one hop neighbour called verifier node, is used for detecting faulty data. Furthermore the verifier node after checking the data, if they are not faulty it forwards them to the sink adding one bit indicating that the data are not faulty.

5.2.4 Remarks

During the attempt we did to categorize all the fault detections we picked, we faced a number of difficulties. The heterogeneity of each approach led as to pick very generalized characteristics from each one and group them accordingly in order to be possible to evaluate them after.

The calculation-based approaches are characterized by the model which is the most important component of the approach. Nevertheless, the models can vary a lot between them but by this grouping we can perform an efficient evaluation and derive useful conclusions about them. The protocol-based approaches are defined as approaches that detect a fault with a certain protocol, most of the times based on message exchanging. The message exchange pattern in these approaches are critical for the performance of the approach. The hybrid approaches were not able to be described from the previous two categories because they use both a model and protocol during their function, to this end we define another category named hybrid. The design of a hybrid may be more complex because it may include the cooperation of a model with a protocol or a number of models with a protocol.

Chapter 6

Evaluation of Fault Detection Approaches

In this chapter we introduce a set of evaluation criteria which will be used to evaluate the picked fault detection approaches and infer conclusions about them. Next we present the tables with the obtained data of the picked approaches. We have to mention that some papers do not provide all the needed data and some records in the table are not completed. Finally we did some conclusions regarding the energy-efficiency and the performance of the fault detection approaches, by using the data from the table and the evaluation criteria.

6.1 Evaluation Criteria

Fault detection approaches in WSNs most of the times are designed for specific applications. We evaluate a fault detection approach from two aspects: application-independent and application-dependent criteria. The former considers mostly characteristics of of fault detection approaches as algorithms, while the latter considers the characteristics which are more related to the application.

A fault detection approach can be evaluated as an algorithm, from its computation complexity, correctness, robustness and etc. Mahapatro et al. [17] analyse several terminologies, including correctness, completeness, consistency, latency, computational complexity,communication complexity, diagnosability, detection accuracy, false alarm rate. In this thesis, we adopt *detection accuracy, false alarm rate, computational complexity and communication complexity.*. The definitions of these application-independent criteria are as follows.

- *Detection Accuracy(DA)*: The ratio of the number of faulty nodes detected to the actual number of the actual number of faulty nodes in the network.
- *False Alarm Rate(FAR)*: The ratio of the number of fault-free nodes detected to the actual number of of fault-free nodes in the network.
- *Computational Complexity (COMP)*: The amount of computing resources needed by a fault detection algorithm.
- *Communication Complexity (COMM)*: Total number of messages exchanged in a WSN used for detecting faults.

Except application-independent criteria, there are several application-dependent criteria. Such criteria are the Fault Type(FTYPE), which is what types of fault, the approach is able to detect. Some other criteria that we adopt are Message Exchange Pattern(MEP), Communication Range(CR), which are mentioned in information collection section and also Assumptions(ASMP), Calculation Method(CM), and Output Range(OR) which are mentioned in decision making.

6.2 Evaluation Data

In table 6.1 we describe some general characteristics of each fault detection approach. The first column contains the reference of the paper we examine. In second column, *C* stands for the category. The next column contains the application scenario, the next column describes the fault type. Next, *MEP* stands for the message exchange pattern, *MC* stands for the content of the message, *CR* stands for the communication range, *CM* stands for the calculation method and *CM* stands for the the output range.

Paper	C	APP Scenario	FTYPE	MEP	MC	CR	CM	OR
[39]	Ca	detection of faulty reading	I	active probing	hello, location, readings	neighbourhood	weighted voting	itself
[24]	Ca	improving the detection accuracy	F	passive observing	detection status	neighbourhood	based on [20]	itself
[45] & [46]	Ca	fault detection accuracy	I	passive observing	N/A	pair wise	ARX model, reference free	itself
[26]	Ca	Fault detection	F,C	passive observing	22 metrics, readings	BS-nodes	temporal/spatial correlation in system metrics	BS range
[43]	Ca	Fault detection in BSN	I	passive observing	alarm messages, calculating data	pair-wise	posterior probability	itself
[30]	Ca	Data correction on a central server	I	passive observing	readings	central server	Hierarchical Bayesian space time modelling	itself
[49]	Ca	Data correction on a fusion center	I	passive observing	readings	fusion center	Bayesian Networks, Neyman Person test	itself

Paper	**C**	**APP Scenario**	**FTYPE**	**MEP**	**MC**	**CR**	**CM**	**OR**
[40]	Ca	trusty sensor selection	I	passive observing	readings	fusion center	hierarchical Bayesian space time modelling	itself
[44]	Ca	binary decision evaluation	I	passive observing	binary decisions	fusion center	threshold test	itself
[47]	Ca	centralized data fault detection	I	passive observing	end-to-end time, readings	node-sink	centralized naive Bayes Detector	network, itself
[50]	Ca	data fault detection	I	passive observing	readings	node-sink	expectation maximization algorithm	itself
[34]	Ca	minimize network burden	F,C	passive observing	readings	BS-nodes	failure knowledge library	itself
[51]	Ca	anomaly detection in medical WSNs	F,I,C	passive observing	readings	BS-nodes	decision tree J48	itself
[28]	P	keep connection within cluster	F	passive observing	hello,location, energy,ID fail report	cluster	simple judgement	itself
[25]	P	detecting faulty leaf	F	active probing	test packet	cluster	comparing test results	itself
[27]	P	detecting crashed nodes in a cluster	F	passive observing	notify packet	cluster	notify packet declares the node is alive	itself

Paper	C	APP Scenario	FTYPE	MEP	MC	CR	CM	OR
[33]	P	link error or crash detection	F,I	active probing	hello packet	within CHs	active probing message mechanism	itself
[29]	P	fault detection	F,C	active probing	scenario data input/output	observer-BS, BS-nodes	time constraints check	itself
[48]	Hy	linear dynamic	I	passive observing	readings	arbitrary group	group testing, Kalman filtering	itself
[32]	P	detecting faults in large scale WSNs	F,C	passive observing	readings, useful data	one hop	fault detection based on FSM	itself
[52]	P	detecting faulty nodes	I	passive observing	fault status useful data	neighbours	threshold test decision dissemination	itself
[35]	Hy	Identification of local status	F,C	active probing	readings, test results	neighbours	threshold test, evaluation from neighbours	itself
[42]	Hy	Data correction application	I	passive observing	readings	neighbours	group voting	itself
[53]	Hy	detecting faulty sensors	F,I,C	passive observing	uniform distribution	neighbours	threshold test	itself
[31]	Hy	fault detection with a fusion tree	F,C	active probing	beacons, local evidence	parent children	naive Bayesian classifier, evidence fusion	parent, children
[14]	Hy	event monitoring	F,I,C	passive observing	readings, fault reports	neighbours	neighbour voting	itself
[41]	Hy	data correction	I	passive observing	readings,	neighbours	MRF,correlation with neighbours	itself

Paper	C	APP Scenario	FTYPE	MEP	MC	CR	CM	OR
[20]	Hy	locate the faulty sensors	F	passive observing	readings, test results	neighbours	neighbour readings comparison	itself
[38]	Hy	ensure accuracy and reliability of sensor data	F,I	active probing	readings	neighbours	group voting, time series analysis	itself
[54]	Hy	collaborative online change detection	I	N/A	sufficient statistics	leader node-collaborators	Markov linear state	cluster
[55]	Hy	data fault detection	I	passive observing	readings	pair-wise	HMM, threshold test	itself
[36]	Hy	data fault detection	I	active probing	readings, requests ,data	neighbours	verifier node mechanism	one hop neighbours

Table 6.1: Fault Detection Approaches for WSNs

In table 6.3 we list the application independent data that we obtained through the scientific papers we picked. In the fourth column of the table we have the detection accuracy, in the next columns we have the false alarm rate and in the last column we have the communication complexity. We need to mention that we did not use the COMP(computation complexity) for two reasons. First, because we consider that the energy-efficiency is defined more from the COMM(communication complexity) than the COMP. Even if an algorithm is long and complex the consumed energy is much lower than the consumed energy from the transmission of a message. Second, because the papers do not provide all the required details for calculating the specific criterion and it was impractical. Another information we want to add is that the COMM is the number of the messages occurred from a fault detection approach. Describing this criterion is complex and we used a specific notation described in 6.2

H	Header
M	Number of the parents
m	Number of the children
N	Number of the nodes in the WSN
n	Number of the nodes in the neighbourhood
CH	Number of the Cluster Heads
reading	The sensor measurement
ff	Number of fault free nodes
fn	Number of faulty nodes[33] [14]
D	Depth of the tree [25]
LN	Leaf nodes [25]
OB	Number of observer nodes [29]
int	integer variable
array	array variable
bool	boolean variable
double	double variable
char	char variable

Table 6.2: Notation for the attributes we used from fault-detection approaches

Paper	C	APP Scenario	DA	FAR	COMM
[39]	Ca	detection of faulty reading	> 0.8	< 0.2	$N(H) + Nn(H) + Nn(H + double) + Nn(H + bool)$
[24]	Ca	improving the detection accuracy	0.992	0.3	$Nn(H + reading)$
[45] & [46]	Ca	fault detection accuracy	> 0.9	N/A	$N(H + reading)$
[26]	Ca	Fault detection	N/A	N/A	$Nn(H + reading)$
[43]	Ca	Fault detection in BSN	0.73	N/A	$N(H + reading) + N(H + array)$
[30]	Ca	Data correction on a central server	> 0.68	< 0.02	$N(H + reading)$
[49]	Ca	Data correction on a fusion center	0.7	0.11	$N(H + reading)$
[40]	Ca	trusty sensor selection	0.974	0.008	$N(H + reading)$
[44]	Ca	binary decision evaluation	N/A	< 0.05	$ND(H + reading)$
[47]	Ca	centralized data fault detection	> 0.7	0.05	$N(H + reading)$
[50]	Ca	data fault detection	0.95	0.05	$N(H + reading)$
[34]	Ca	minimize network burden	> 0.9 for F 0.75 for C	> 0.35 for F 0.3 for C	$N(H + reading)$
[51]	Ca	anomaly detection in medical WSNs	1	0.048	$N(H + reading)$
[28]	P	keep connection within cluster	N/A	N/A	$Nn(H + double) + M(H + double) + 2fn(H)$
[25]	P	detecting faulty leaf	> 0.9	N/A	$2 * D * LN * (H)$
[27]	P	detecting crashed nodes in a cluster	N/A	N/A	$2N(H) + fn(H)$
[33]	P	link error, crash detection	0.9	N/A	$2CH(H)$
[29]	P	fault detection	N/A	N/A	$2OB(H + double) + 2N(H + double)$
[32]	P	detecting faults in large scale networks	0.9	0.1	$N(H + double)$
[52]	P	detecting faulty nodes	>0.91	< 0.1	$Nn(H + boolean)$
[48]	Hy	linear dynamic systems	> 0.85	< 0.02	N/A

Paper	C	APP Scenario	DA	FAR	COMM
[20]	Hy	locate the faulty sensors	0.955	0.025	$Nn(H + reading)$ $+3Nn(H + double)$
[35]	Hy	Identification of local status	N/A	N/A	$N(H + double)$ $+Nn(H + bool)$ $+Nn(H + bool)$
[42]	Hy	Data correction application	> 0.9	< 0.1	$Nn(H + reading)$ $+Nn(H + bool)$
[53]	Hy	detecting faulty sensors	1	N/A	$Nn(H + double)$
[38]	Hy	ensure accuracy and reliability of sensor data	< 0.8	< 0.3	$Nn(H + reading)$ $+N(H + reading)$
[31]	Hy	fault detection with a fusion tree	> 0.76	< 0.24	$fn(n(2H) + (LN(2H)+$ $m(H + char)$ $+M(h) + (H)))$
[14]	Hy	event monitoring	> 0.8	< 0.38	$Nn(H + reading) + fn(H)$
[41]	Hy	data correction	> 0.96	0.0038	$Nn(H + reading)$
[54]	Hy	collaborative online change detection	0.88	N/A	$N(H + reading)$
[55]	Hy	data fault detection	> 0.99	< 0.02	$N(H + reading)$
[36]	Hy	data fault detection	> 0.99	0.0052	$2(H) + 3N(H)+$ $N(H + reading)$ $+4N(H + double)$

Table 6.3: Application-independent criteria for fault detection approaches in WSNs

In table 6.4 we list in which of the picked approaches adopt the assumptions described in table 5.2. In the next section we evaluate the approaches and we see which of the assumptions can impact the performance of the approaches.

Paper	C	FU_1	FU_2	FU_3	IN_1	IN_2	CO_1	CO_2	CO_3	FA_1	FA_2	FA_3
[39]	Ca	Y	N	N	N	Y	Y	Y	N	Y	Y	N
[24]	Ca	Y	N	N	N	Y	Y	N	Y	Y	Y	N
[45] & [46]	Ca	Y	N	N	N	Y	Y	N	N	Y	Y	N
[26]	Ca	Y	N	N	N/A	N/A	Y	Y	N	Y	Y	N
[43]	Ca	Y	N	N	N	N	N/A	Y	N	Y	Y	N
[30]	Ca	Y	N	N	Y	Y	N/A	N	N	Y	Y	N
[49]	Ca	Y	N	N	N	Y	N/A	N	N	Y	Y	N
[40]	Ca	Y	N	N	N	Y	Y	N	N	N	N	N
[44]	Ca	Y	N	N	N	N	Y	N	N	Y	Y	N
[47]	Ca	Y	N	N	N	N	N	N	N	Y	Y	N
[50]	Ca	Y	N	N	N	N	N	Y	N	Y	Y	N
[34]	Ca	Y	N	N	N	Y	N/A	N	N	Y	Y	N
[51]	Ca	Y	N	N	N	Y	N	N	N	Y	Y	N
[28]	P	Y	N	N	Y	N	Y	Y	N	Y	Y	N
[25]	P	Y	N	N	N/A	N/A	N	Y	N	Y	Y	N
[27]	P	Y	Y	N	N	N	N/A	Y	N	Y	Y	N
[33]	P	Y	N	N	N	Y	N	Y	Y	Y	Y	N
[29]	P	Y	N	N	N	N	N	N	N	Y	Y	N
[32]	P	Y	N	N	N	N/A	N	N	N	Y	Y	N
[52]	P	Y	N	N	N	Y	N/A	N	Y	Y	Y	N
[48]	Hy	Y	N	N	N	Y	Y	N	N	Y	Y	N
[20]	Hy	Y	N	N	N	Y	Y	N	Y	Y	Y	N
[35]	Hy	Y	N	N	N	Y	N	Y	N	Y	Y	N
[42]	Hy	Y	N	N	N	Y	Y	Y	N	Y	Y	N
[53]	Hy	Y	N	N	N	Y	N	N	N	N	N	Y
[38]	Hy	Y	N	N	N	N	N	N	N	Y	Y	N
[31]	Hy	Y	N	N	N	N/A	N	Y	N	Y	Y	N
[14]	Hy	Y	N	N	N	Y	Y	N	N	Y	Y	N
[41]	Hy	Y	N	N	N	Y	N/A	N	N	Y	Y	N
[54]	Hy	Y	N	N	N	Y	N/A	Y	N	Y	Y	N
[55]	Hy	Y	N	N	N	Y	N/A	N	N	Y	Y	N
[36]	Hy	Y	N	N	N	Y	N	Y	Y	Y	Y	N

Table 6.4: Application-dependent criteria for fault detection approaches in WSNs

6.3 Discussion

Over this section we review the data from the previous tables in order to interpret them and derive useful conclusions regarding the energy-efficiency and the performance of the picked fault detection approaches. We structure this section in communicational and computational performance. The communicational performance includes the energy-efficiency evaluation in combination with other evaluation criteria. We call it communicational because, as it was stated before, the communicating messages affect in a great degree the energy-efficiency. The computational performance includes an evaluation taking into the DA and FAR of each of the picked approaches in combination with other evaluating criteria. In other words, we evaluate the energy-efficiency and the performance of the picked approaches, taking into account the evaluation criteria which were mentioned in the previous section.

6.3.1 Communicational Performance

In this part we are going to examine how several criteria are able to affect the energy-efficiency of a fault-detection approach. The main objective is to derive observations and have a better view of the whole image of energy-efficiency. In order to evaluate the energy-efficiency, we have to focus on the COMM criterion which characterize the energy-efficiency of a fault detection approach. The COMM criterion is mentioned in table 6.3.

Energy-Efficiency over categories

Many of the calculation-based approaches seem to have as computational complexity(COMM) the number of the nodes in the WSN. In such fault detection approaches, the number of the messages are reduced and appear to be more energy-efficient than the other two categories. The protocol-based approaches, regarding the COMM criterion, appear to be less energy-efficient than the calculation-based approaches but more than the hybrid approaches. Most of the protocol-based approaches are based on message exchange mechanisms, this is the factor which increase the energy consumption and consequently reduces the energy-efficiency. The hybrid approaches have the highest number of messages, which can be explained by the fact that a hybrid approach is like executing a combination of a calculation-based and a protocol based approach, thus the energy consumption is increased. The conclusions we made here are the following:

- The calculation based approaches consume the lowest amount of energy over the three categories
- The protocol-based approaches consume more amount of energy than the calculation based but less that hybrid approaches
- the Hybrid approaches consume the highest amount of energy over the three categories

Relation between Energy-Efficiency and Topology

Here we investigate the topology impact on the energy-efficiency of a fault detection approach. We focus on the fault detection approaches which have a specific topology and it is applied the assumption (ASMP-CO-2). The topologies we examine are distinguished in *cluster-based* and *tree-based*. First, for the cluster-based approaches, we cannot say that we observe any similarities to the energy-efficiency, that's why the COMM criterion vary among the approaches. Nevertheless, we can say that the cluster based is less energy efficient than the tree-based. It seems that the tree-based topology requires less messages to complete a fault detection, thus it consumes less energy. The conclusions we can extract here are the following:

- There is no similarities in COMM criterion between same topologies
- Tree-based fault detection approaches may be more energy efficient than the cluster-based.

Relation between Energy-Efficiency and MEP

In this part we present how the MEP affects the energy-efficiency of the fault detection approaches. The message exchange patterns we consider in this thesis are mentioned in chapter 4.2, namely active-probing and passive-observing. The former can be described as a request-reply form and the latter as one-way broadcast. It is obvious that approaches which use active probing as MEP consume more energy. The reason is also obvious, because they require more messages to complete a fault detection and consequently, more energy.

- The fault detection approaches which use passive observing are more energy efficient.

Relation between Energy-Efficiency and CM

Here we examine how the CM of a fault detection approach can impact its energy-efficiency. It is very challenging to group the CMs and evaluate them as a group because maybe they use some basic principles from fundamental mathematical models but in general they are different. We tried to be coherent and we use the categories of *Bayesian Network*, *Message Coordination Protocol*, and *Threshold Test*. The Bayesian network CMs are the ones which use basic principles from the Bayesian network model. The Message Coordination Protocol are CMs based on messages e.g. periodic test with "Hello-IAmALive" messages. The the last category of CMs is based on threshold tests to detect a fault. We have to mention that we do not include the CMs from hybrid approaches, as the evaluating data refer to a combination of of CMs and not only one.

Regarding the CMs based on Bayesian networks, they appear to be the most energy efficient. Many of them are based purely on a mathematical model and the result is calculated locally. The fact that there is no need of extra messages makes these CMs energy-efficient. The threshold-test CMs are consuming more energy than the previous category. The reason for the increased energy consumption here is that the threshold tests are disseminated after being calculated and need extra information to be calculated. The message coordination protocol CMs consume more energy than the previous two categories. The fact that they function with messages increase in great degree the energy consumption and makes them the least energy efficient between the three categories. The derived conclusions here are:

- CMs based on Bayesian Network are the most energy-efficient
- CMs based on Threshold Tests consume more energy than th CMs based on Bayesian Network but less than CMs based Message Exchange Protocols
- CMs based on Message Coordination Protocols are the least energy-efficient.

6.3.2 Computational Performance

This part presents the fault detection approaches computational performance under a series of different evaluation criteria. The performance is characterized by the detection accuracy(DA) and the false alarm rate(FAR).

Performance over categories

Here we examine how each category perform, regarding the table 6.3. For the category of calculation-based fault-detection approaches, if we focus on the DA rate, we can see that is above 0.7 and the FAR is bellow 0.2 in overall, except the case of [34], which have a false alarm rate more than 0.35, for functional faults and 0.3, for communicational faults. The next category, protocol based appear to have DA at least 0.9 or higher and FAR at least 0.1 or lower. The metrics in Hybrid category are above 0.76, for the DA and bellow 0.38, for FAR.

In order to have a more comprehensive view over the categories, we calculated the mean values of the DA and FAR criteria for every category. In figure 6.1 we can see the mean values of the fault detection approaches we picked over the three different categories, namely calculation-based, protocol-based and hybrid. What we can see is, that the difference in detection accuracy between calculation-based and hybrid approaches, is very low. Another observation is, that the protocol-based approaches have slightly higher detection accuracy. According to the figure 6.1, the protocol-based category seems to have a FAR value of 0.1, which is slightly lower in compare with calculation-based and hybrid which have 0.137 and 0.121 accordingly. Regarding to the evaluating results we did the following observations:

- The protocol-based approaches may perform better in overall, as they have the highest DA and the lowest FAR, over the three categories
- The calculation-based and hybrid approaches performance are very close, although the approaches from the latter category seem to perform slightly better.

Relation between Performance and Topology

The objective here is to examine if the topology dependent criterion(ASMP-CO-2) can affect the performance of a fault detection approach. The topologies we consider again are the *cluster-based* and the *tree-based.* The figure 6.2 depicts the mean values regarding the performance of the cluster-based and the tree-based topologies. As we can see there is no tremendous difference to the DA, although the approaches using tree-based topology seem to present slightly higher DA but also little more FAR. The conclusions we derived are the following :

- The fault detection approaches using the describing topologies do not seem to have great differences between them

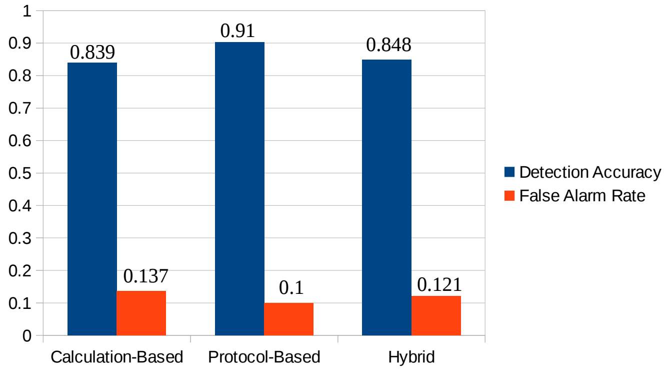

Figure 6.1: The mean values of the detection accuracy and false alarm rate over the propose categorization

- The approaches using tree-based topology seem to have slightly higher DA but little more FAR

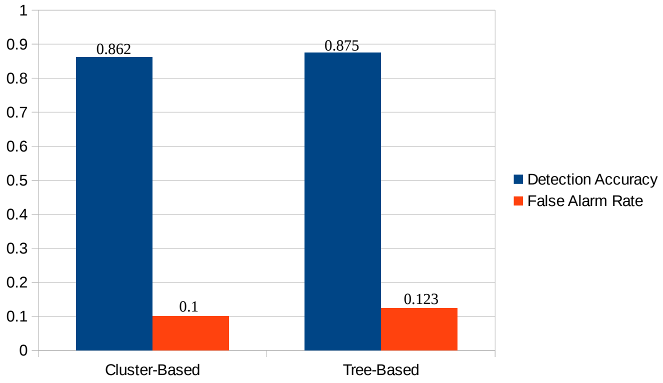

Figure 6.2: The mean values of the detection accuracy and false alarm rate regarding the approaches using cluster-based and tree-based topology

Relation between Performance and MEP

Another interesting observation, is how the MEP affects the performance of the picked approaches. More specifically, we calculated the mean value DA and FAR of the approaches which use passive observing and active probing accordingly. In figure 6.3 we can see results. What we can infer taking into account the results is:

- Using the passive observing MEP we have slightly lower DA, however using the same MEP we have lower FAR

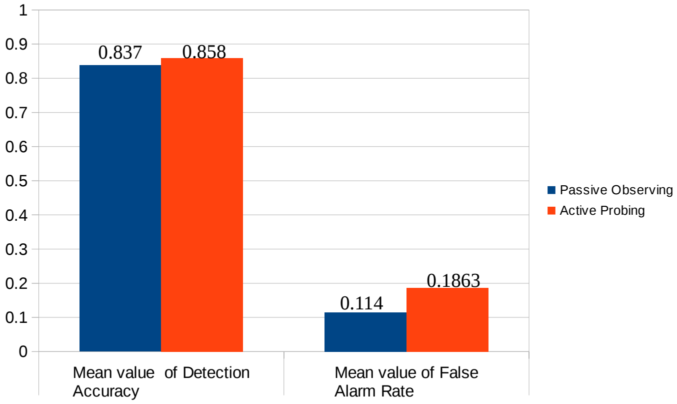

Figure 6.3: The mean values of the detection accuracy and false alarm rate regarding the message exchange patterns

Relation between Performance and CM

Here we see how a CM can affect the performance of a fault detection approach. The challenges for grouping the CMs are the same with the section which we examine the relationship between the energy-efficiency and the CM. The categories we picked are also the same. We can see in figure 6.4 that the threshold test CMs have the highest accuracy and the CMs based Bayesian network have the lowest DA. Regarding the FAR the Bayesian network CMs have the lowest and the CMs based on message coordination protocols have the highest. What we can infer here is:

- The CMs based on the threshold tests have the highest DA
- The CMs based on Bayesian networks have the lowest FAR

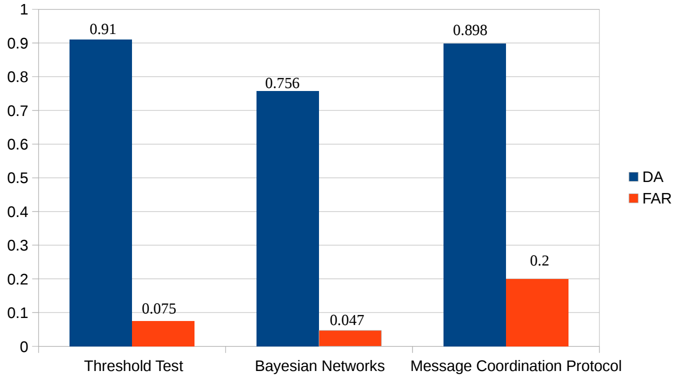

Figure 6.4: The mean values of the detection accuracy and false alarm rate regarding the calculation methods

Relation between Performance and Correlation

In the following figure 6.5, we examine how the correlation of the sensor readings can affect the performance of an approach. According to the results, it is clear that when we use the assumption ASMP-IN-2, we can achieve higher results in detection accuracy, although the false alarm rate is slightly increased also.

- A fault detection approach which takes advantage of the correlation of the sensor readings, may have higher detection accuracy but the false alarm rate may be also higher.

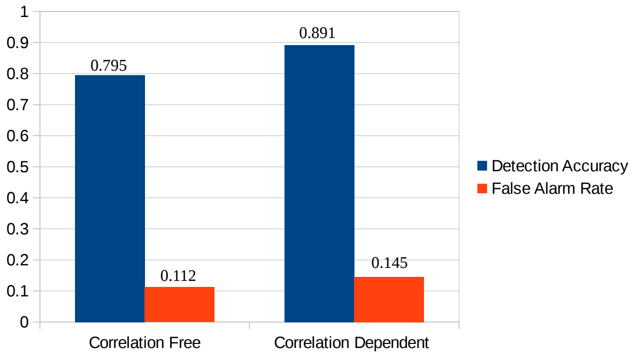

Figure 6.5: The mean values of the detection accuracy and false alarm rate regarding the ASMP-IN-1 criterion

Chapter 7

Design Guideline

Designing a fault detection method for WSNs is a complex procedure. Since the WSNs applications are dependent on the requirements and to the deployment environment, each fault detection method should be designed regarding application specific criteria. In this chapter we provide a set of advices which can be useful for a designer of a fault detection method. The guideline we provide here is structured in two sections, the first section, computational performance, includes advices regarding the energy efficiency. The second section, computational performance, provides advises regarding the performance of the fault detection regarding the DA and the FAR.

7.1 Communicational Performance Design Guideline

As we mentioned before the communicational performance refers mainly to energy-efficiency. The energy-efficiency is the main consideration of a designer when he designs an application in WSNs. The fact that if a sensor node runs out of energy becomes useless, makes energy-efficiency the first priority.

Regarding the categorization we are proposing in this the thesis, a calculation-based fault detection is more likely to consume less energy. Over the selected approaches the topologies we examined are the cluster-based and the tree-based. An advice regarding the topology is that between the two mentioned topologies the tree-based may consume less energy regarding our results. If a designer has the option to choose between the two MEPs, the passive observing is the more energy efficient one. The CMs we distinguish over the picked approaches are the *threshold-test, Bayesian networks and message coordination protocol* the Bayesian Network appear to be more energy efficient over the others.

7.2 Computational Performance Design Guideline

In this section we are proposing advices regarding the DA and FAR. The energy-efficiency is not the only thing that matters in a fault detection method. In some cases the DA and FAR are equally important as the energy-efficiency. For example, a fault detection approach for a military application which monitors the battlefield, it is highly important have high computational performance.

The first advice in this section is that, the protocol-based approaches appear to have better performance, as they have higher DA and lower FAR than the other two categories. The using topology in fault detection approaches cannot offer tremendous changes but between the cluster-based and tree-based topologies, the former may have slightly lower FAR and the latter little more DA. Regarding the option of the MEP, by using the passive observing we may have lower FAR but the for slightly higher DA we have to use the active probing. According to the selected fault detection approaches, the CM which offer the higher DA is the threshold test and the one which offer lower FAR is the Bayesian networks. We have to mention that the CMs we consider are the same as the previous section. Finally if the design is based on the correlation of the sensor readings, it will have higher DA but slightly higher FAR.

Chapter 8

Conclusion

This thesis is about analysis of fault detection method in Wireless Sensor Networks. The fact that we are not dealing with a specific problem raises up several challenges. One of them is to specify the fault type classification we should use and we decided that the proposed classification provides wider domain and is more practical for the scope of this thesis. The identification of the fault detection framework and its division into phases required thorough research and deep understanding of the topic. Another challenge was the evaluation part. The fact that some papers from the literature do not provide the required information or they provide it in a very abstract way, made our attempt to obtain the required information difficult. The contribution list can be summarized as following:

- Proposal of our own fault type classification
- Proposal of fault detection framework in WSNs
- Proposal of an evaluation criteria set for fault detection approaches in WSNs.
- Evaluation of selected fault detection approaches from performance and energy-efficiency perspective.
- Proposal of a design guideline for a fault detection method

First we researched the existing literature to find scientific papers including a fault detection approach for WSNs. We proposed a fault type classification which has a wide domain and can be used in the context of research or industry as well as well, not considering security threats as they are outside of our research domain We also proposed a framework for fault detection methods for WSNs which can provide better understanding of concept improve it. We used the proposed evaluation criteria in order to evaluate the picked fault detection approaches and we saw how they affect a fault detection approach in terms of energy-efficiency and performance. The results are used to propose a design guideline and help a designer develop a fault detection approach for WSNs according the desired requirements.

To the best of our knowledge, this is the first work that analyse the fault detection framework in WSNs and evaluated a great amount of approaches in order to use the results for providing a design guideline. This made this thesis complete and coherent.

Future Work

This thesis is based on data obtained from scientific papers and the overall level is theoretical. The way to add practical part and be more coherent is implementing the picked fault detections approaches, and use a network simulator for checking the performance and the energy efficiency of each approach. An alternative future direction of this thesis would be doing experiments on sensor nodes used in real-life situations. Nevertheless, implementing all the algorithms proposed by the scientific papers would demand a lot of effort and could be cumbersome. Many of the papers describe in an abstract way the particular algorithms or in worse cases they omit to mention the technical details. Nonetheless, the practical implementation of the algorithms and the subsequent empirical analysis would result in a more complete research, with indisputable outcomes.

Bibliography

[1] C. Buratti, A. Conti, D. Dardari, and R. Verdone, “An overview on wireless sensor networks technology and evolution.,” *Sensors (Basel, Switzerland)*, vol. 9, pp. 6869–96, 2009.

[2] V. Jolly and S. Latifi, “Comprehensive study of routing management in wireless sensor networks - part - ii.,” in *ICWN*, pp. 49–62, 2006.

[3] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, “Mac essentials for wireless sensor networks,” *Communications Surveys Tutorials, IEEE*, vol. 12, pp. 222–248, Second 2010.

[4] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, “A taxonomy of wireless micro-sensor network models,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, pp. 28–36, Apr. 2002.

[5] D. Pescovitz, “Brainy buildings conserve energy,” *UC Berkeley College of Engineering Lab Notes*, 2001.

[6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA ’02, (New York, NY, USA), pp. 88–97, ACM, 2002.

[7] M. K. Bob Fornaro and K. Angione, “Tiny sensor-based computers could help track wildlife,” *North Carolina State University*, 2003.

[8] A. Milenković, C. Otto, and E. Jovanov, “Wireless sensor networks for personal health monitoring: Issues and an implementation,” *Comput. Commun.*, vol. 29, pp. 2521–2533, Aug. 2006.

[9] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Computer Networks (Elsevier)*, vol. 46, pp. 605–634, 2004.

[10] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, pp. 2787–2805, Oct. 2010.

[11] A. Tanenbaum and M. van Steen, *Distributed systems: principles and paradigms.* Pearson Prentice Hall, 2007.

[12] F. F. Ben-Gal I., Ruggeri F. and K. R., "Bayesian Networks, Encyclopedia of Statistics in Quality and Reliability,," 2007.

[13] C. M. Grinstead and L. J. Snell, *Grinstead and Snell's Introduction to Probability.* American Mathematical Society, version dated 4 july 2006 ed., 2006.

[14] J. Gao, J. Wang, and X. Zhang, "Hmrf-based distributed fault detection for wireless sensor networks," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 640–644, Dec 2012.

[15] M. Yu, H. Mokhtar, and M. Merabti, "A Survey on Fault Management in Wireless Sensor Networks," 2007.

[16] R. Jurdak, X. R. Wang, O. Obst, and P. Valencia, "Chapter 12 Wireless Sensor Network Anomalies : Diagnosis and Detection Strategies," pp. 309–325, 2011.

[17] A. Mahapatro and P. M. Khilar, "Fault Diagnosis in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2000–2026, 2013.

[18] M. Barborak, A. Dahbura, and M. Malek, "The consensus problem in fault-tolerant computing," *ACM Comput. Surv.*, vol. 25, pp. 171–220, June 1993.

[19] A. G.-R. Arsan Munir, Joseph Antoon, "Modeling and analysis of fault detection and fault tolerance in wireless sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 5, 2014.

[20] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks - DIWANS '06*, p. 65, 2006.

[21] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 2, pp. 902–913 vol. 2, March 2005.

[22] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. Sen. Netw.*, vol. 5, pp. 25:1–25:29, June 2009.

[23] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, pp. 11–33, Jan 2004.

[24] P. Jiang, "A new method for node fault detection in wireless sensor networks.," *Sensors (Basel, Switzerland)*, vol. 9, pp. 1282–94, Jan. 2009.

[25] A. Taleb, J. Mathew, and D. Pradhan, "Fault diagnosis in multi layered de bruijn based architectures for sensor networks," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 456–461, March 2010.

[26] X. Miao, K. Liu, Y. He, Y. Liu, and D. Papadias, "Agnostic diagnosis: Discovering silent failures in wireless sensor networks," in *INFOCOM, 2011 Proceedings IEEE*, pp. 1548–1556, April 2011.

[27] L. Karim and N. Nasser, "Energy efficient and fault tolerant routing protocol for mobile sensor network," in *Communications (ICC), 2011 IEEE International Conference on*, pp. 1–5, June 2011.

[28] G. Venkataraman, S. Emmanuel, and S. Thambipillai, "A cluster-based approach to fault detection and recovery in wireless sensor networks," in *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on*, pp. 35–39, Oct 2007.

[29] C. Ramassamy, H. Fouchal, P. Hunel, and N. Vidot, "A pragmatic testing approach for wireless sensor networks," in *Proceedings of the 6th ACM Workshop on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '10, (New York, NY, USA), pp. 55–61, ACM, 2010.

[30] E. W. Dereszynski and T. G. Dietterich, "Spatiotemporal Models for Data-Anomaly Detection in Dynamic Environmental Monitoring Campaigns," *ACM Transactions on Sensor Networks*, vol. 8, pp. 1–36, Aug. 2011.

[31] Q. Ma, K. Liu, X. Miao, and Y. Liu, "Sherlock is Around: Detecting Network Failures with Local Evidence Fusion," in *INFOCOM 2012*, Mar. 2012.

[32] K. Liu, Q. Ma, X. Zhao, and Y. Liu, "Self-diagnosis for large scale wireless sensor networks," in *In Proceedings of IEEE INFOCOM*, 2011.

[33] S. Bhatti, J. Xu, and M. Memon, "Energy-aware fault-tolerant clustering scheme for target tracking wireless sensor networks," in *Wireless Communication Systems (ISWCS), 2010 7th International Symposium on*, pp. 531–535, Sept 2010.

[34] J. Nie, H. Ma, and L. Mo, "Passive diagnosis for wsns using data traces," in *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*, pp. 273–280, May 2012.

[35] E. Khazaei, A. Barati, and A. Movaghar, "Improvement of fault detection in wireless sensor networks," in *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on*, vol. 4, pp. 644–646, Aug 2009.

[36] A. R. M. Kamal, C. Bleakley, and S. Dobson, "Packet-level attestation (pla): A framework for in-network sensor data reliability," *ACM Trans. Sen. Netw.*, vol. 9, pp. 19:1–19:28, Apr. 2013.

[37] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Trans. Sen. Netw.*, vol. 6, pp. 23:1–23:39, June 2010.

[38] T. A. Nguyen, D. Bucur, M. Aiello, and K. Tei, "Applying time series analysis and neighbourhood voting in a decentralised approach for fault detection and classification in WSNs," in *Proceedings of the Fourth Symposium on Information and Communication Technology - SoICT '13*, (New York, New York, USA), pp. 234–241, ACM Press, Dec. 2013.

[39] D. De, "A distributed algorithm for localization error detection-correction, use in in-network faulty reading detection: Applicability in long-thin wireless sensor networks," in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pp. 1–6, April 2009.

[40] K. Ni and G. Pottie, "Sensor network data fault detection with maximum a posteriori selection and bayesian modeling," *ACM Trans. Sen. Netw.*, vol. 8, pp. 23:1–23:21, Aug. 2012.

[41] A. Farruggia and S. Vitabile, "A novel approach for faulty sensor detection and data correction in wireless sensor network," in *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2013 Eighth International Conference on*, pp. 36–42, Oct 2013.

[42] S. Dobson and D. Hughes, "An Error-free Data Collection Method Exploiting Hierarchical Physical Models of Wireless Sensor Networks,"

[43] D.-J. Kim and B. Prabhakaran, "Motion fault detection and isolation in Body Sensor Networks," *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 147–155, Mar. 2011.

[44] H.-T. Pai, "Reliability-based adaptive distributed classification in wireless sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, pp. 4543–4552, Nov 2010.

[45] C. Lo, J. P. Lynch, and M. Liu, “Pair-wise reference-free fault detection in wireless sensor networks,” in *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, IPSN ’12, (New York, NY, USA), pp. 117–118, ACM, 2012.

[46] C. Lo, M. Liu, and J. Lynch, “Distributive model-based sensor fault diagnosis in wireless sensor networks,” in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, pp. 313–314, May 2013.

[47] B. C. Lau, E. W. Ma, and T. W. Chow, “Probabilistic fault detector for wireless sensor network,” *Expert Systems with Applications*, vol. 41, pp. 3703–3711, June 2014.

[48] C. Lo, M. Liu, J. Lynch, and A. Gilbert, “Efficient sensor fault detection using combinatorial group testing,” in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, pp. 199–206, May 2013.

[49] K. Ni and G. Pottie, “Bayesian selection of non-faulty sensors,” in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pp. 616–620, June 2007.

[50] E. M. Shakshuki, X. Xing, and T. R. Sheltami, “An intelligent agent for fault reconnaissance in sensor networks,” in *Proceedings of the 11th International Conference on Information Integration and Web-based Applications &Amp; Services*, iiWAS ’09, (New York, NY, USA), pp. 139–146, ACM, 2009.

[51] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, “Sensor fault and patient anomaly detection and classification in medical wireless sensor networks,” *2013 IEEE International Conference on Communications (ICC)*, pp. 4373–4378, June 2013.

[52] M.-H. Lee and Y.-H. Choi, “Fault detection of wireless sensor networks,” *Comput. Commun.*, vol. 31, pp. 3469–3475, Sept. 2008.

[53] P. Zhuang, D. Wang, and Y. Shang, “Distributed Faulty Sensor Detection,” *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pp. 1–6, Nov. 2009.

[54] H. Snoussi and C. Richard, “Wsn06-5: Distributed bayesian fault diagnosis in collaborative wireless sensor networks.,” in *Global Telecommunications Conference, 2006. GLOBECOM ’06. IEEE*, pp. 1–6, Nov 2006.

[55] E. Warriach, T. A. Nguyen, M. Aiello, and K. Tei, “A hybrid fault detection approach for context-aware wireless sensor networks,” in *Mobile Adhoc*

and Sensor Systems (MASS), 2012 IEEE 9th International Conference on, pp. 281–289, Oct 2012.