Wireless Sensor Networks for Industrial Health Assessment Based on a Random Forest Approach

Wiem Elghazel

SPIM

Thèse de Doctorat

UFC

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE FRANCHE-COMTÉ

Wireless Sensor Networks for Industrial Health Assessment Based on a Random Forest Approach

WIEM ELGHAZEL

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UFC UNIVERSITÉ DE FRANCHE-COMTÉ

THÈSE présentée par

WIEM ELGHAZEL

pour obtenir le

Grade de Docteur de
l'Université de Franche-Comté

Spécialité : **Automatique**

Wireless Sensor Networks for Industrial Health Assessment Based on a Random Forest Approach

Soutenue publiquement le 9 Décembre 2015 devant le Jury composé de :

CONGDUC PHAM	Rapporteur	Professeur à l'Université de Pau, France
FRANÇOIS PÉRÈS	Rapporteur	Professeur à l'ENI Tarbes, France
JEAN-CLAUDE BOCQUET	Examinateur	Professeur Émérite à Centrale-Supélec Paris, France
MAHMOUD BARHAMGI	Examinateur	Maître de Conférences à l'Université Claude Bernard à Lyon, France
VINCENT BOUILLET	Invité industriel	Ingénieur chez Alstom Power à Grenoble, France
JACQUES BAHI	Directeur de thèse	Professeur à l'Université de Franche-Comté
NOUREDDINE ZERHOUNI	Directeur de thèse	Professeur à l'ENSMM de Besançon, France
KAMAL MEDJAHER	Co-encadrant	Maître de Conférences- HDR à l'ENSMM de Besançon, France
MOURAD HAKEM	Co-encadrant	Maître de Conférences à l'Université de Franche-Comté, France
CHRISTOPHE GUYEUX	Co-encadrant	Professeur à l'Université de Franche-Comté, France

REMERCIEMENTS

Je remercie tout particulièrement l'équipe encadrante : le Professeur Noureddine Zerhouni, le Professeur Jacques Bahi, le Professeur Christophe Gyueux, Monsieur Kamel Medjaher et Monsieur Mourad Hakem pour m'avoir proposé de travailler avec eux sur la thématique du « Prognostics and Health Management using Wireless Sensor Networks ». Je les remercie pour leur confiance, leur soutien, leurs conseils. Ce travail n'aurait pas pu être accompli sans leurs critiques constructives.

Je remercie également tous les personnels de l'ENSMM, du département AS2M et du département DISC pour l'excellente ambiance de travail, les échanges et l'aide précieuse apportée tout au long de la réalisation de ces travaux de recherche, d'encadrement et d'enseignement.

Enfin, je remercie ma famille et mes amis pour leur soutien indéfectible tout au long de mon parcours et lors de la préparation et rédaction de ce mémoire. Je remercie tout particulièrement mon fiancé Slim et mes parents pour leur soutien et leurs encouragements quotidiens. Sans eux, je n'aurais sans doute pas accompli les travaux présentés dans ce mémoire.

SOMMAIRE

TABLE DES FIGURES

LISTE DES TABLES

I

State of the art and Issues

1

GENERAL INTRODUCTION

During their life cycle, and from the moment of their deployment, industrial systems are subjected to failures. A failure might be irreversible or have undesirable outcomes with consequences varying from minor to severe. Generally speaking, system failure can lead to :

- An interruption of service. The service can be critical and even a brief stoppage could degrade the quality of response or prevent the system from providing the intended output.
- Money loss, like the case of grounded airplanes where the loss counts in millions of dollars. Extra costs can also be related to system repair.
- Jeopardizing the security of the users in case of severe outcomes (fires, explosions, gas leaks, etc.), not to mention endangering the environment.

An illustration of the severe consequences is given in Figure [1] 1.1.

In the 21^{st} century alone, many notable industrial disasters took place. For instance, in the defense industry, an incident occurred on July 11^{th} 2011 in Zygi, Cyprus (see Figure 1.1(c)). 98 containers of explosives self-detonated after being stored for two and a half years in the sun. The explosion killed 13 people and injured 62 others. Hundreds of nearby buildings and the island's largest power station were damaged. The costs of this explosion were valued at 10% of the country's economy, which is equal to 3.822 American billion dollars.

On July 6^{th} 2013, a derailment of a runway train took place in Quebec (see Figure 1.1(a)). A combination of neglected defective locomotive, poor maintenance, driver error, flawed operating procedures, weak regulatory oversight, and lack of safety redundancy caused the death of 42 victims and 5 other people were presumed dead. More than 30 buildings were destroyed, while 36 were demolished due to contamination. 115 businesses were either destroyed, displaced, or rendered inaccessible. Claims to local insurers were estimated at a total of $ 50 million.

The Deepwater Horizon oil spill (see Figure 1.1(b)), which happened in Gulf of Mexico from April 20^{th} until July 15^{th} 2010, killed 11 individuals and 143 spill-exposure cases were reported to the local hospital. This was caused by a wellhead blowout, and the well was officially sealed only on September 19^{th} 2010. The spill volume was estimated at 4.9 million barrels spreading on an area of 6,500 to 176,100 Km^2. This cost the company about $ 65.425 billion between fines and settlements.

1. Images courtesy (a,d) wikipedia.org, (b) nypost.com, (c) CyprusNewsReport.com

TransAsia Airways Flight 235 (Figure 1.1(d)) is a domestic flight that crashed on February 4^{th} 2015 shortly after takeoff from Taipei Songshan Airport. The aircraft's right engine triggered an alarm just 37 seconds after takeoff. Whereas the crew reported a flame-out, data showed one of the engines had in fact been moved into idle mode. Soon the right engine failed to produce enough thrust for its rotating propeller, lapsing into auto-feathering. A restart was attempted, but the aircraft crashed 72 seconds later. From 58 passengers and crew members on board, 43 were killed. The remaining 15, in addition to 2 victims on ground, were injured. Right after the crash, 100 flights were canceled. So far, this has caused the company around US$ 475,000 as fees of compensation to the families of the deceased.

(a) Lac-Mégantic rail disaster, Quebec.

(b) Deepwater Horizon oil spill, Gulf of Mexico

(c) Evangelos Florakis Naval Base explosion, Cyprus.

(d) TransAsia Airways Flight 235, Taiwan.

FIGURE 1.1 – Examples of failure outcomes.

The human casualties cost the plants the trust of their clients, which reduces dramatically their equities and incomes. Production stoppage and repair durations also generate countless money loss and expensive fees. From this context, it is important to monitor the system, assess its health, and plan maintenance activities preferably before the system fails.

Maintenance is an important activity in industry. It is performed either to revive a machine/component, or to prevent it from breaking down, and aims for increasing system availability, readiness and enhancing safety. Different strategies have evolved through time, bringing maintenance to its current state. This evolution was due to the increasing demand of reliability in industry. Nowadays, plants are required to avoid shutdowns while offering safety, availability, and reliability, all while reducing the costs.

The first form of maintenance is the corrective one. In this strategy, actions are only taken when the system breaks and no longer can perform the intended tasks. These actions could be equipment repair or replacement. The corrective maintenance reduces the cost of manpower related to maintenance actions. However, plants cannot afford to undergo breakdowns. In fact, sudden shutdowns cost money due to the interruption of production and time due to repair activities. A component failure, in some cases, can lead to a secondary failure, worsening the situation and elevating the costs.

As a remedy to these issues, maintenance became a periodic activity. Domain experts rely on their knowledge and the observation of upcoming events to set time intervals in which the components are inspected and replaced if needed. This preventive maintenance is especially adopted by transportation and nuclear plants [Hu et al., 2012]. A time-based maintenance will encompass any action that aims at adjusting signs of deterioration, in a way that stretches the time interval until a failure occurs. This strategy reduces the occurrence of system breakdowns and increases reliability. The main drawback of preventive maintenance is the fact that it is performed regardless of the machine's condition. In other words, industrials have to hire domain experts in order to set intervals for maintenance. Sometimes, these intervals are irrelevant as the machine can be in a healthy state and this will cost extra and avoidable fees. Besides, even with periodic maintenance and inspections, random failures still occur. This is why Condition Based Maintenance (CBM) was proposed and developed in early nineties [Heng et al., 2009].

CBM is based on real-time observations. It is an online approach that assesses machine's health through condition measurements. As any maintenance strategy, CBM aims at increasing the system reliability and availability while reducing maintenance costs. The benefits of this particular strategy include avoiding unnecessary maintenance tasks and costs, as well as not interrupting normal machine operations [Heng et al., 2009]. CBM decreases the number of maintenance operations and causes a reduction of human error influences.

A new maintenance has recently emerged : Predictive Maintenance (PM). It predicts the system health in the future, based on the current condition, and defines the needed maintenance activities accordingly. This way, the system is only taken out of service if a direct evidence exists that deterioration has actually taken place. This increases maintenance efficiency and productivity on one hand, and decreases maintenance support costs and logistics footprints on the other.

Figure 1.2 summarizes this evolution of maintenance strategies through time.

In order to shift from traditional maintenance strategies to CBM and PM, extra tasks are required. These tasks encompass system surveillance, data analysis and modeling, and decision making support system. This scientific approach is called Prognostics and Health Management (PHM).

Over the past years, research in PHM field has gained a great deal of attention. Prognostic models are developed in an attempt to predict the Remaining Useful Life (RUL) of machinery before failure takes place. This is done following the steps described in Figure 1.3.

A reliable prediction activity requires continuous online measurements of the operating conditions of the system under consideration. This information is usually gathered by means of sensor nodes. Sensory data are reported periodically to monitor the critical components. The data packets contain signals that correspond to measurements of mo-

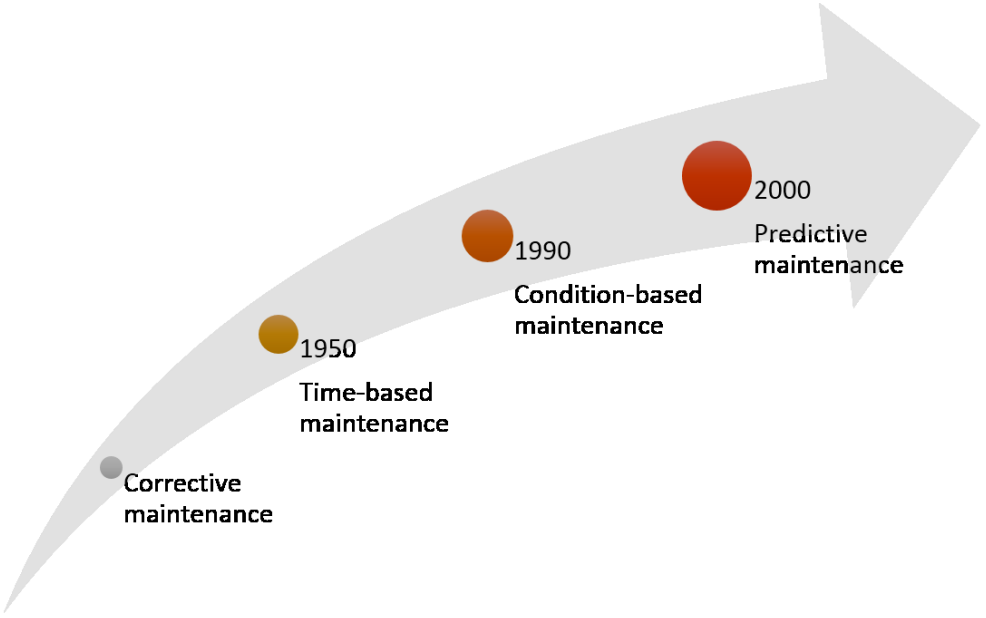


FIGURE 1.2 – History of maintenance strategies.

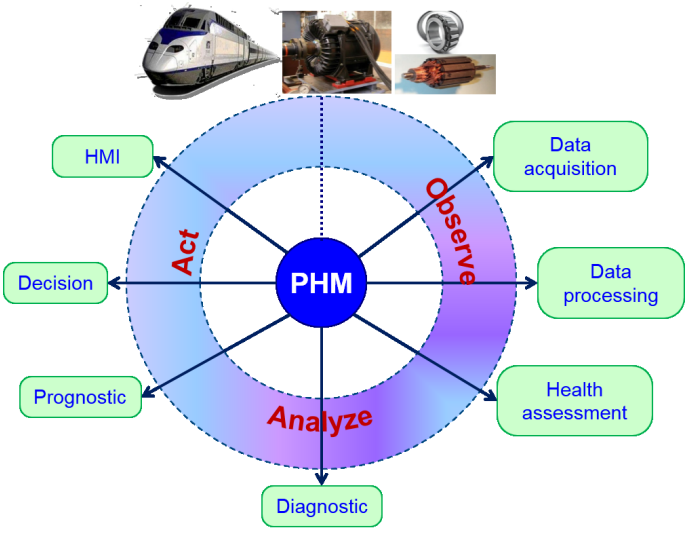


FIGURE 1.3 – PHM general steps.

nitoring parameters such as pressure, temperature, moisture. . .

This information is used for a comprehensive analysis of the system behavior. This comprehension will help build models to assess the health, diagnose the system, and extrapolate the results in the future to estimate the time before failure. After these steps being achieved, the decision making support system will help the user plan the necessary maintenance actions if needed. However, if the prediction model and the provided measurements are not accurate, it is possible that the maintenance activity will be performed either "too soon" or "too late". This is illustrated in Figure 1.4.

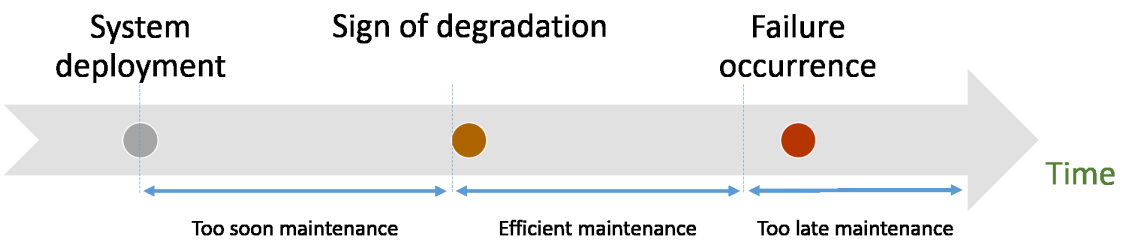


FIGURE 1.4 – Maintenance activity.

1.1/ POSITIONING OF THE RESEARCH

PHM can be performed through one of four main approaches : (i) model-based approach, (ii) knowledge-based models, (iii) data-driven models, or (iv) hybrid models. Model-based approaches rely on a mathematical representation of the component responding to stress. Although the model is reliable, it requires intensive experimentation and verifications and could only be built by a domain expert. Knowledge-based models also explore domain experts' experience to derive models from rules built upon observations. This approach's complexity grows exponentially with the number of the derived rules. Data-driven approaches are used when the first principles of the system operation are too complex to develop an accurate model of physics of failure. Finally, hybrid approaches combine two different approaches among data-driven, model-based, and knowledge-based approaches. This combination will leverage the advantages of each one of the used approaches. However, these models could be computationally expensive as well as complicated to develop.

In data-driven approaches, models are directly derived from condition monitoring data, based on statistical and stochastic learning techniques. These models have a double role : assess current operating conditions and predict the RUL. Neither human expertise nor comprehensive system physics are needed for the model building process.

A data-driven approach transforms raw data provided by the monitoring system into useful information. By means of this information and historical records, a behavioral model is built and predictions can be performed. In this thesis, we develop a data-driven approach for on-line health assessment using the Random Forest algorithm (RF).

To the best of our knowledge, existent research works in PHM perform data collection either by means of individual sensors, or via wired networks of sensor nodes. In some monitoring applications, the deployment of a Wireless Sensor Network (WSN) is mandatory rather than a choice for the following reasons.

– Due to accessibility restrictions, it is sometimes easier to place wireless sensors in the monitoring area, without the constraint of connecting them with wires.
– Some systems are sensitive to extra weight, and eliminating the wires can guarantee better performances.
– We can get closer to the source of the data we need, comparing to when we use wires, it can also become easier to deploy more sensors in the monitoring process, and thus get more precise values.

For precision and feasibility purposes, we consider in this study the case where the nodes communicate their information within a WSN.

Two of the most hot topics related to WSN are energy consumption and network reliability. Energy consumption can be optimized through the choice of topology, the deployment of the routing protocol, and node scheduling. In this thesis we investigate different topologies to illustrate their impact on the quality of data. We also propose a node scheduling scheme that keeps a minimum number of sensor nodes in the active mode while ensuring coverage.
Network reliability solutions aim at improving the network availability (by employing fault tolerance techniques) and minimizing packet loss. However, this does not completely eliminate the risks of loosing information. Our proposed algorithm stores copies of the sensed data into the neighboring nodes memory. These copies are retrieved once the sender fails before delivering the data packet. The data packets arrived at the sink node will later be used for PHM.

A majority of the current research works in PHM study the accuracy of models from the angle of improving prognostics algorithms only. In this thesis, we focus our attention on earlier steps, namely : (i) data acquisition, (ii) data processing, (iii) and health assessment.

The quality of the available data for the PHM process is not always considered in the literature. The solutions proposed in the literature are based upon the assumption of completeness and correctness of data. This assumption, however, is far from being true especially in the case of WSN monitoring. Therefore, we focus our attention on a data-driven approach that can adapt to the change of quality of the monitoring data. More precisely, we use the random forest algorithm for health assessment.

1.2/ RESEARCH QUESTIONS

WSN monitoring is somehow unique in the sense that the sensors are also subjected to failures or energy exhaustion, leading to a change in the network topology. Thus, monitoring quality is variable too and it depends on both time and the location of sensors on/around the device. Our challenges conducting health assessment, while monitoring the system with a WSN, arise as follows :

- Data acquisition step : how is the data being gathered ? Which routing protocol is employed ? Which topology is best ?
- Data processing step : is the data complete ? Is the amount of data at the base station sufficient for health assessment ?
- Health assessment step : In the light of the monitoring constraints, how can we estimate the health state ?

Various strategies can be deployed on the network to achieve fault tolerance or to extend the WSN's lifetime, like nodes scheduling or data aggregation. However, the health assessment process must be compatible with these strategies, and with a device coverage of a changing quality. Another challenge related to this work, is finding an algorithm that adapts to the change in number of sensors.

1.3/ GLOBAL ASSUMPTIONS

Data-driven approaches consist in a machine learning algorithm to understand the degradation mechanism of the system in question. The input of such an algorithm would be sensor measurements describing the degradation of the system over time. A challenging step in a data-driven approach is data gathering. Actually, the quality of sensor data affects the efficiency of the health assessment process. A prior study by system experts is needed to ensure a strong basis for reliable models. The global assumptions considered in this research work are as follows.

1. The optimal number of sensor nodes, their location, and the parameters to be monitored are previously determined by system experts.
2. Data acquisition begins when the system starts functioning and stops only when it completely fails or all sensors have exhausted their energy resources.
3. During data acquisition, no maintenance activity takes place.
4. The degradation develops gradually over time.
5. Data is ready for use.
6. The network used for the monitoring activity is secured.

1.4/ CONTRIBUTIONS OF THE THESIS

WSNs are designed for the purpose of an efficient event detection. They consist of a large number of sensor nodes deployed in a surveillance area to detect the occurrence of possible events. Such an activity necessitates efficiency, which is hard to achieve with the constraints of WSNs. A prerequisite in PHM is to consider that data provided by sensors are either flawless or simply noisy, which is far from the truth. We, therefore, studied some aspects of WSN dependability. We studied different settings of the WSN. We showed that altering the topology has an impact on the quality of data arriving at the base station for health assessment. We also developed a fault tolerant technique, enabling the network to fulfill its intended task even in the presence of faults. Thus, there will be no interruptions in forwarding data for health assessment purposes. A data recovery process is also developed, it aims at reducing data loss by creating copies of packets in non-exhaustive way. A node scheduling scheme is included in the algorithm and it helps to improve the energy consumption.

In this thesis, the use of random forests (RF) is proposed for industrial functioning health assessment, particularly in the context of devices being monitored using a wireless sensor network (WSN). In the offline phase, the algorithm selects the relevant features for the best split in the decision tree, starting from the root until the leaf nodes are reached. This process is repeated until the maximum number of trees is reached. The algorithm, since it is based on random selections, has the advantage of starting from different points (or distributions). Therefore, in the online phase, this can be beneficial for mainly two reasons :

– Health assessment can be performed considering the relevance of all features. A majority vote will then assign the correspondent health class, and this reduces errors.

– If one feature is missing (due to sensor error), the health assessment process will not be interrupted as other trees have a different starting point.

To summarize, the contribution of this research work is performing health assessment with WSN monitoring, render the network more reliable, and improve the quality of predictions by focusing on the early steps of the process. The proposed solution is summarized in Figure 1.4.

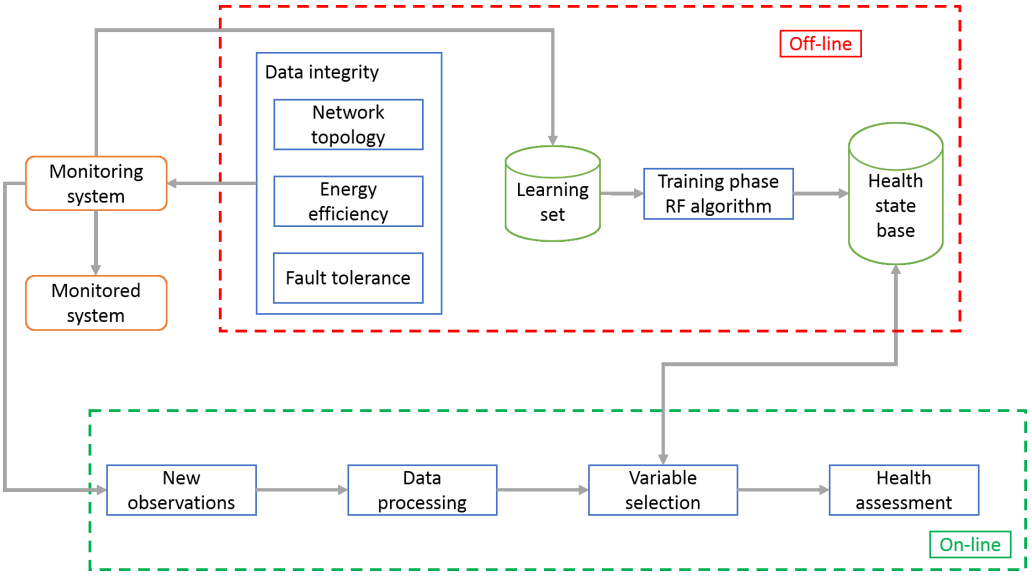


FIGURE 1.5 – The overall scheme of the proposed solution.

1.5/ PUBLICATIONS

1.5.1/ PUBLISHED PAPERS

1. Dependable wireless sensor networks for prognostics and health management : a survey. Annual Conference of the Prognostics and Health Management Society, PHM'14, / Fort Worth - Texas - USA (Volume 68, 2014, Pages 1-15). Wiem Elghazel, Kamal Medjaher, Christophe Guyeux, Mourad Hakem, Noureddine Zerhouni, Jacques Bahi.

2. Random Forests for Industrial Device Functioning Diagnostics Using Wireless Sensor Networks. IEEE AEROSPACE CONFERENCE, 2015. / Big Sky, Montana, USA (2015, Pages 1-9). Wiem Elghazel, Jacques Bahi, Ahmad Farhat, Christophe Guyeux, Mourad Hakem, Kamal Medjaher, Noureddine Zerhouni.

3. Dependability of wireless sensor networks for industrial prognostics and health management. Computers in industry (Volume 68, January 2015, Pages :1-15). Wiem Elghazel, Jacques Bahi, Ahmad Farhat, Christophe Guyeux, Mourad Hakem, Kamal Medjaher, Noureddine Zerhouni.

1.5.2/ SUBMITTED PAPERS

1. Resiliency in Distributed Sensor Networks for Prognostic and Health Management of the Monitoring Targets. Jacques Bahi, Wiem Elghazel, Christophe Guyeux, Mohammad Haddad, Mourad Hakem, Kamal Medjaher, and Noureddine Zerhouni.
   Submitted to the computer journal, Oxford.
2. Reliable diagnostics using wireless sensor networks. Wiem Elghazel, Jacques Bahi, Christophe Guyeux, Mourad Hakem, Kamal Medjaher, Noureddine Zerhouni.
   Submitted to Journal Européen des Systèmes Automatisés.
3. Health assessment with incomplete data using wireless sensor networks. Wiem Elghazel, Jacques Bahi, Christophe Guyeux, Mourad Hakem, Kamal Medjaher, Noureddine Zerhouni.
   Submitted to IEEE Sensors Journal.
4. Data recovery and energy efficiency in distributed wireless sensor networks. Jacques Bahi, Wiem Elghazel, Christophe Guyeux, Mohammad Haddad, Mourad Hakem, Kamal Medjaher, and Noureddine Zerhouni.
   Submitted to INFOCOM 2016.

1.6/ THESIS OUTLINE

The remainder of this thesis is organized as follows :

Chapter 2 presents an overview of wireless sensor networks and their advantages. It also states the weaknesses of this type of networks, to better understand what threatens the quality of service. Some of the solutions to reach network's dependability reported in the literature are listed.

Chapter 3 compares the different maintenance strategies that evolved through time. Condition-based maintenance is put under the spot light and useful definitions regarding prognostics and health management are given. Various prognostics approaches are explained and discussed while highlighting each of their advantages and drawbacks, in a way that justifies the choice of data-driven approaches. This chapter also discusses the challenges of wireless sensor networks monitoring in the context of industrial prognostics and health management. Finally, this chapter emphasizes what needs to be tackled for the purpose of obtaining good results.

Chapter 4 presents a distributed algorithm for resiliency in wireless sensor networks. The algorithm has two main goals : (1) reducing energy consumption in the network and (2) recovering data loss. The algorithm consists in a node scheduling mechanism. This mechanism not only ensures that only one sensor node is active per area, but also that each area is covered at all times (as long as there still exist alive nodes). An area is delimited by a distance equal to the nodes' radio range. Two different mechanisms for data recovery are proposed, depending on whether memory constraints exist or not.

Chapter 5 proposes the use of the random forest algorithm for health assessment. The algorithm is used to identify the health state of an industrial device when the monitoring parameters are incomplete. This method is validated through computer simulations while varying the data collection process.

Chapter 6 summarizes the work that was achieved and enumerates the contributions of this thesis. It also discusses the limitations and the perspectives.

2

An overview of wireless sensor networks

Wireless Sensor Networks (WSN) are a new technology that appeared in the nineties. The emergence of WSN is the result of the advancement in low cost wireless communication and micro electro mechanical systems [Akyildiz et al., 2002]. These networks became wild spread thanks to their easy deployment and their low cost. In the past, getting data transferred from one unit to another requested expensive wiring. Nowadays, we can place nodes almost anywhere, having no wiring contrasts, with any geographic or emplacement problems, and this is due to the minimal batteries feeding the nodes. The sensors' small dimensions give them the advantage to be used even discreetly. These advantages allowed WSN to be deployed in various fields, such as : agriculture, medicine, home automation, military, etc.

Typically, a WSN is composed of few base stations (one in most cases) and hundreds (or thousands) of sensor nodes. The main role of these devices is to monitor physical or environmental conditions, and cooperate to deliver the sensed data to a base station called the sink.

WSN are composed of very small sensors with very limited and nonrenewable energy. In order to preserve this energy, network throughput has to be low. Furthermore, as all wireless networks, WSN are not very secure. A node can easily be hacked or failed, either due to energy limitation or to network mobility. Moreover, they do not dispose with a predefined infrastructure and this fact highlights the importance of the chosen routing protocol. An adequate protocol should ensure a reliable communication. This reliability means reducing data loss, fastening communication, minimizing energy consumption, and other standards.

WSN are event-based systems that rely on the collective effort of several sensor nodes [Akan et al., 2005]. This offers the network greater accuracy, larger coverage area, and the possibility to extract localized features. The network extends the computational capability to physical environments that human beings cannot reach. Generally, sensor nodes are not located far from each other (since they function at a low frequency). As a consequence, it is highly possible that many nodes sense the same data. And if all this redundant information is routed through the network, useful energy will be dissipated in vain.

In this chapter, a general overview of WSN is provided. This encompasses their advantages, drawbacks, and the solutions implemented to improve their performance.

2.1/ DEFINITIONS

A sensor node is a tiny device having the capability of sensing new events, computing the sensed and received values, and communicating information. Thus, the network can be deployed to monitor physical and environmental phenomena such as temperature, vibrations, light, humidity, etc.

As shown in Figure 2.1, the node is equipped with a sensor, a data processing unit, a memory space, a radio range, and a battery.

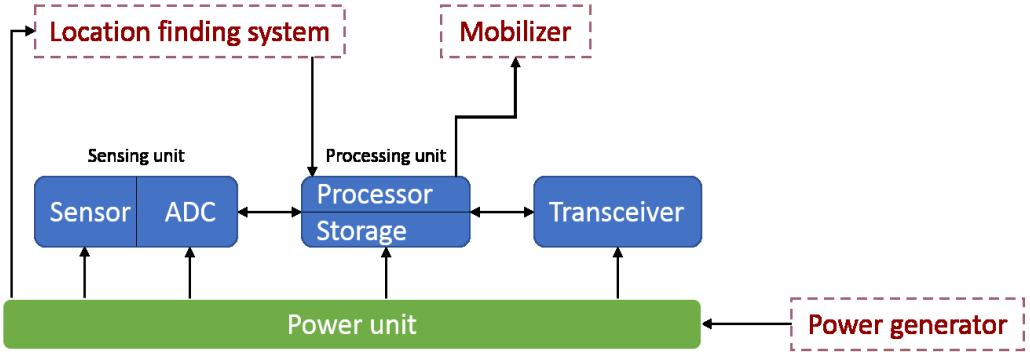


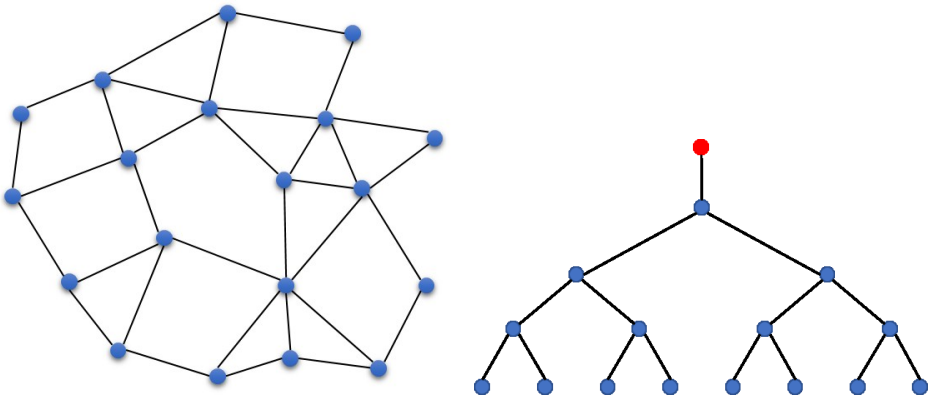
FIGURE 2.1 – Components of a sensor node

Ad hoc is the Latin for "for this", and it means a non-generalizable solution designed for a specific problem or task.
A WSN is an ad hoc network as it does not rely on a pre-determined infrastructure. Each node participates in routing the packets by forwarding data for other nodes determined dynamically on the basis of network connectivity.

WSNs can be either heterogeneous or homogeneous [Li et al., 2011]. In the latter, all nodes have the same role and characteristics. In the former, nodes have different roles : some nodes simply sense and forward information while others aggregate data, manage their area, perform computations... Consequently, some of the nodes can be equipped with higher energy, longer radio range...

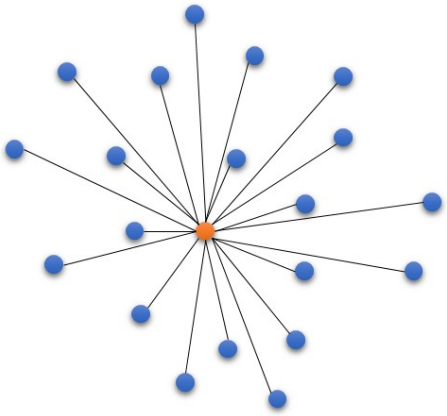
There are different settings for a WSN, which is generally dynamic as radio range and network connectivity change over time [Li et al., 2011]. The most used WSN models are hierarchical, distributed, centralized, or decentralized (Figure 2.2).

In the distributed topologies, there is no central node managing the network (or a region of it). They consist in a collection of nodes having equal roles. Therefore, no aspect of hierarchy is considered. Before the network starts running, no prior infrastructure is imposed ; each node discovers its surrounding area and decides which node (or nodes) to communicate with. This decision usually relies on the radio range and the transfer distance. With this topology, it is easy to add new sensor nodes to the network ; as they only need to establish links with their neighbors. Distributed topologies render the network's maintenance an easy task : should a node fail, its neighbors will establish new links with other nodes within their sensing range, and the network will continue to work normally. Nevertheless, this topology may result in extra costs related to energy consumption and security codes.

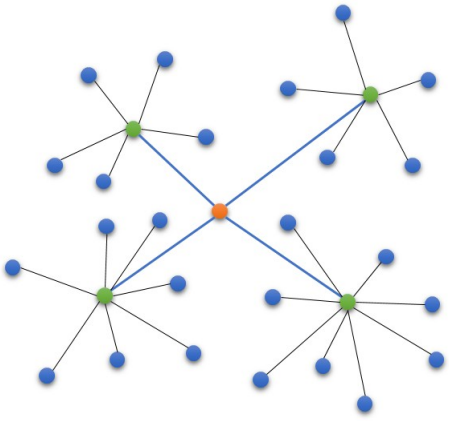


(a) Distributed topology

(b) Hierarchical topology



(c) Centralized topology



(d) Decentralized topology

FIGURE 2.2 – Network topologies

Sensor nodes can be organized in several levels, making a hierarchical topology (or a tree topology). The root represents level 0 and there is no level above. Sensor nodes from two adjacent levels are connected in an end to end manner. The hierarchical model can be seen as three different layers : (1) the core layer (the root), which is optimized for availability and performance, (2) the distribution layer, which implements policies and forwards messages, and (3) the access layer (the leaf nodes), which represents the access point to the network. Hierarchical WSN have the advantage of being scalable. The presence of different levels makes the network more manageable and simplifies the task of isolating and detecting faults. Unfortunately, when the network spans on a great area, maintenance can be an issue. If a parent node fails, its children can no longer communicate with other nodes in the network. Apart from leaf nodes, any node failure in the network will result in disjoint sub-trees.

One of the easiest topologies to design and implement is the centralized topology (also called star topology). All the sensor nodes have the simple task of sensing new information and forwarding it to a central node where all the data processing will be proceeded with. Adding new nodes to the topology is a simple task ; the central node is known and only a connection with the new sensor needs to be established. One of the major problems of this topology is that it presents a single point of failure. If a problem occurs at the central node, the whole network becomes paralyzed : when a new event is detected, the

data packet cannot be forwarded nor processed. The network loses purpose once it loses its central node. In addition to this, the network does not ensure speed. In fact, increasing the network's density will lead to higher traffic. An important number of forwarded packets towards the same point may result in a traffic jam and packets collision, known in this case as the bottleneck problem.

Decentralized topologies can be seen as a combination of the distributed and the centralized topologies. The network is divided into regions (or clusters) which are locally managed by a central node (called the Cluster Head CH). This topology is best known for offering a reasonable compromise between energy consumption and Quality of Service (QoS). Adding new nodes and managing the topology would be as simplified as in the centralized topologies. In the same time, the congestion problem is reduced and the network no longer has a single point of failure. However, once a CH fails, the corresponding area is no longer monitored.

Choosing a network setting depends on the target application (mainly its goals and limits), and the routing protocol will be developed accordingly. The protocol stack of sensor networks is composed of five different layers [Akyildiz et al., 2002], which are shown in Figure 2.3.

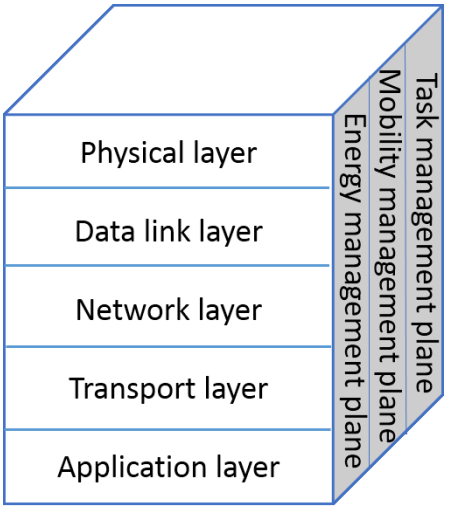


FIGURE 2.3 – Protocol stack for WSN.

1. **Physical layer :** it defines the means of transmitting data packets after being converted into raw data bits, that are suitable for transmission over the communication medium. Mainly, this layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation, and data encryption.
2. **Data link layer :** It is charged of data stream multiplexing, data frame creation, medium access, and error control to provide reliable transmission. it is also charged of the creation of the network infrastructure, transferring data, and fairly and efficiently sharing the communication resources between sensor nodes, in order to achieve good network performance in terms of energy consumption, network throughput, and delivery latency. It is also responsible for error control of transmission data.
3. **Network layer :** it is responsible for routing the data from the source nodes until the sink. It also ensures inter-networking with external networks, where the sink node

can be used as a gateway. It commands and controls the system, forwards data packets, and takes charge of routing between intermediate routers.

4. **Transport layer :** it is responsible for end-to-end data delivery between sensor nodes and the sink. Due to energy, computation, and storage constraints, traditional transport protocols cannot be applied directly to WSNs. This layer also provides other services such as multiplexing, reliability, flow control, congestion avoidance...
5. **Application layer :** it includes a variety of protocols that perform various sensor applications, such as query dissemination, node localization, time synchronization, network security, etc. It can be defined as the user interface. It displays messages in a human recognizable and understandable format.

Due to their low cost, their easy deployment, and their capacity to extract localized features, WSN are widely spread. Nowadays, we can find them nearly in all monitoring applications. However, to the best of our knowledge, the use of this technology is not reported in the literature of Prognostics and Health Management (PHM). Nevertheless, in some industrial applications, the WSN monitoring can be mandatory. This is further explained in the next section.

2.2/ WIRELESS SENSOR NETWORKS FOR INDUSTRIAL MONITORING

The monitoring activity begins with the simple question : how many sensor nodes do we need ?
First of all, the number of sensors depends on the number of monitored parameters. The sensors have different types, so for each of the monitored parameter, we need at least one sensor for the monitoring activity. Secondly, for large systems, more than one strategic place can be found to extract localized values for the parameter under consideration. Therefore, situations where more than one sensor node is needed are very common.

Once we have determined the number of sensors, the second question we ask ourselves is : how is the collected data routed to the base station ?
Most of the encountered research works use individual sensors for data collection. This means that each sensor transfers its sensed data to the base station (BS) separately. Once at the BS, the data packets are processed and fusion algorithms are deployed to extract correlations.

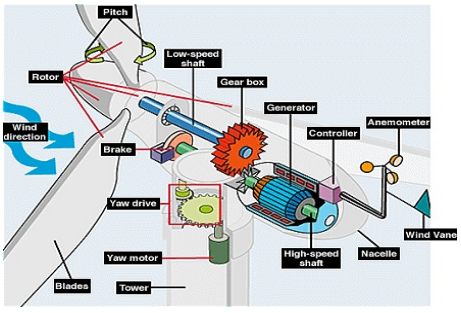
As long as the number of sensors is limited, wiring should not be a problem. But as their number grows higher, connecting the sensors individually to the BS starts to turn into a complicated task. An alternative solution would be connecting the sensors in a hop by hop manner until reaching the final destination (which is th BS). A hop is a portion of the path between the source (sender) and the destination. In a hop by hop communication, the packets are sent from one node to the other until reaching the BS. Therefore, only the nodes that are the closest to the BS are directly wired to it. This should reduce the costs and renders wiring less complicated.

However, even with this solution, wiring can still remain a complicated task. Here for example, we take the case of wind turbines shown in Figure[1] 2.4. A wind farm usually

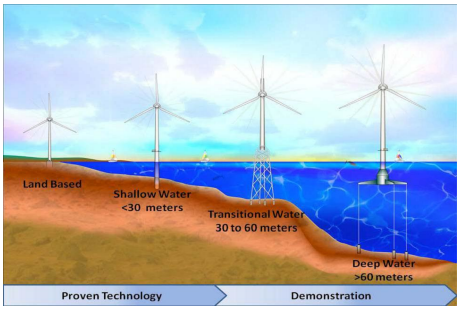
1. Image courtesy : boem.gov

contains a number of wind turbines that can be over one hundred in some cases. Connecting the sensors monitoring each of the wind turbines via wires is not that easy, especially when we face the facts. Wind turbines should be spaced by about 6 to 10 times the diameter of the rotor. Modern wind turbines have diameters varying from 40 to 90 meters. So, in best case scenario, connecting two sensors in adjacent wind turbines would require a wire of 240 meters long, at least ! Connecting each of the wind turbines directly to the base station is clearly worse, and performing localized processing will add the constraint of access, like the case of offshore wind farm in Figure 2.4(c).
Putting this aside, let us have a look at the wind turbine design in Figure 2.4(b). We can see that the generator component is composed of several parts in constant rotation. Adding wires to the equation would clearly complicate things even more.

(a) A field of wind turbines.



(b) Components of a horizontal-axis wind turbine.



(c) Wind turbines set offshore.

FIGURE 2.4 – An example of wind turbines.

The drawbacks of wired networks can also be illustrated by the example of aircraft monitoring. In Figure 2.5 are shown two examples of sensor networks on a Boeing 777 [2], and on an Airbus [3].

In a study conducted by the Sahara project [4], it is stated that an Airbus A380 has 500 miles of cable with a total weight that exceeds 3 tons. The use of WSNs would save at least hundreds of pounds, and thus reduces the consumption of fuel.

Implementing a WSN has a number of advantages that we can summarize as : (i) giving an easier access to extract localized features, (ii) granting access in case of geographical difficulties, (iii) reducing the complexity of manufacturing and associated hazards, and (iv)

2. Image courtesy : Institute of fluid-flow machinery, Polish academy of science
3. Image courtesy : High performance composites- Article : Structural health monitoring : Composites get smart
4. http ://www.beanair.com/success-stories/sahara-2-project

(a) WSN on a Boeing 777. (b) WSN on an Airbus.

FIGURE 2.5 – airplanes.

eliminating faults related to wired communication (such as delays).
Unfortunately, a WSN also has limitations which are reviewed hereafter.

2.3/ SHORTCOMINGS OF A WSN

WSNs are designed for the purpose of an efficient event detection. They consist of a large number of sensor nodes deployed in a surveillance area to detect the occurrence of new events. Such an activity necessitates efficiency, which is hard to achieve with the constraints of WSNs. These limitations are detailed in the following.

2.3.1/ RESOURCES

Available energy is a big limitation to WSN capabilities. The sensors are small sized devices, which results tiny batteries as energy supply. Moreover, the nodes are often deployed in hostile environments (mountains, enemy territory...). So once deployed, they cannot be recharged [Carman et al., 2000].

The added security code has an important impact on the available energy for normal network tasks. Processing security functions (encryption, decryption, signing data, verifications), transmitting security related data (vectors for encryption/decryption), and securing storage (cryptographic key) necessitate extra power, which is critical for WSNs [Carman et al., 2000, Walters et al., 2007].

In addition to this, the deployed memory space for a sensor node is very limited. The storage space is shared between the communication protocol and the security code. The size of the latter has then to be limited to a minimum [Walters et al., 2007].

Buffering space in sensor nodes is also limited. This will lead to packet loss with the increase in traffic flow towards the sink node. In fact, a data packet cannot be held by the sensor for a long period with new packets coming in. In the case of a high traffic flow, all the nodes will attempt to « get rid » of the old messages in order to make space to the new ones, by forwarding them to the next level. Therefore, the area around the sink tends to be quickly congested as all sensor nodes tend to forward the captured data to the sink.

2.3.2/ COMMUNICATION

Wireless communication is known to be unreliable and it adds to the network's vulnerabilities. The absence of physical connections can result in :

- **Channel errors :** Due to noise in the communication channels, the arrival of wrong signals may occur at the recipient.
- **Missing links :** Route updates, interference in the radio channels, energy exhaustion... lead up to invalid or missing links between the sensors and consequently to packet drop.
- **Network congestion :** Heavy traffic is reflected by a dense packet exchange in the network, and a concurrent access may present itself at certain regions. When a node carries so much data, its QoS degrades as this leads to packet collision, packet loss, transmission delays...
- **Communication latency :** Multi-hop routing and node processing both lead to a great latency. Latency is the time elapsed between packet sending and packet reception. When transmission errors require retransmissions, extra delays are to take into consideration. One of the major drawbacks of latency is that it makes synchronization among nodes hard to achieve.

Most WSNs are deployed in harsh environment conditions and/or are exposed to adversary attacks. This emphasizes the likelihood of physical attacks, which can cause permanent (even irreversible) damage to the hardware. Since the network is managed remotely, the sensor nodes can be left unattended for a long period of time. It is therefore difficult to detect physical tampering or perform regular maintenance, and thus, the network would remain unable to fulfill the intended tasks [Walters et al., 2007].

Routing solutions in WSNs avoid central management point as it results in a single point of failure. This complicates the synchronization among nodes, and lowers packet delivery rate. Ensuring synchronization among nodes necessitates access to extra memory space to improve the communication protocol, consuming more resources.

2.3.3/ COVERAGE AND LIFETIME OPTIMIZATION

Considering all the limitations mentioned above, it is not easy for the network to always fulfill the intended tasks. Reliability and efficiency of WSNs are dependent on key issues, which are enumerated in the following.

2.3.3.1/ COVERAGE

Sensor nodes have a short radio range and they collaborate to cover a given surveillance area. At the network setup phase, it is crucial to ensure that the network is configured in a way that it covers all the area [Tian et al., 2005]. The coverage problem arises as : how to ensure that, at any time, any zone in the network is covered by at least one sensor node ?

[Zorbas et al., 2007] presented B{GOP}, a centralized coverage algorithm for WSNs. The algorithm proposes sensor candidate and avoids double-coverage depending on the coverage status of the corresponding field.

[Wang et al., 2003] presented a protocol that can dynamically configure a network to achieve guaranteed degrees of coverage and connectivity. They gave a proof that sensing coverage range does not need to be more than half the connectivity range in the network. Thus, their protocol helps preserve energy while maintaining coverage in the network.

In [Hefeeda et al., 2010], a general coverage algorithm, which considers the network connectivity, is presented. The proposed protocol, called Probabilistic Coverage Protocol (PCP), works for the common disk sensing model as well as probabilistic sensing model. To support probabilistic sensing models, the authors introduce the notion of probabilistic coverage of a target area with a given threshold θ, which means that an area is considered covered if the probability of sensing an event occurring at any point in the area is at least θ. They prove the correctness of the protocol and provide bounds on its convergence time and message complexity.

2.3.3.2/ AWAKE NODES *vs* SLEEPING NODES

In order to prolong the network's lifetime, a possible solution is to keep a minimum number of sensor nodes in active mode. As WSNs rely on nodes density in the sensing and communicating processes, it is very likely that some nodes will not be needed. If a reliable node can forward data packets toward the sink, its neighbors can switch to idle state temporarily.

Lifetime optimization using knowledge about the dynamics of stochastic events has been studied in [He et al., 2012b]. The authors presented the interactions between periodic scheduling and coordinated sleep for both synchronous and asynchronous dense static sensor network. They show that the event dynamics can be exploited for significant energy savings, by putting the sensors on a periodic on/off schedule.

In [He et al., 2012d], the authors design a polynomial-time distributed algorithm for maximizing the lifetime of the network. They proved that the lifetime attained by their algorithm approximates the maximum possible lifetime within a logarithmic approximation factor.

The authors in [Kasbekar et al., 2011a] leverage prediction to prolong the network life time, by exploiting temporal-spatial correlations among the data sensed by different sensor nodes. Based on Gaussian Process, the authors formulate the issue as a minimum weight sub-modular set cover problem and propose a centralized and a distributed truncated greedy algorithms (TGA and DTGA). They prove that these algorithms obtain the same set cover.

As sensor nodes periodically go to sleep, they need to be awake when they are requested to. This is done by the transmission of wake-up messages towards a target sensor. However, if the message is not received at the right moment, data packets will be dropped. This will cost the network extra energy due to packet retransmission [Ye et al., 2003, Gallais et al., 2006, Bahi et al., 2011].

2.3.3.3/ WEAR-OUT EFFECT

In WSN, if the wear-out failures are not taken into consideration during the execution of the involved application, some nodes may age much faster than the others and become the reliability bottleneck for the network, thus significantly reducing the system's service li-

fetime. In the literature, this problem has been formulated and studied in various ways. For instance, prior work [He et al., 2012b, He et al., 2012d, Kasbekar et al., 2011a] in lifetime reliability assumes node's failure rates to be independent of their usage times. While this assumption can be accepted for memoryless soft failures, it is obviously inaccurate for the wear-out-related fail-silent (a faulty node does not produce any output) and fail-stop (no node recovery) failures, because the sensor node's lifetime reliability will gradually decrease over time.

To cope with this problem, a distributed self-stabilizing and wear-out-aware algorithm is presented in [Bahi et al., 2013]. This algorithm seeks to build resiliency by maintaining a necessary set of working nodes and replacing failed ones when needed. The proposed protocol is able to increase the lifetime of wireless sensor networks, especially when the reliabilities of sensor nodes are expected to decrease due to use and wear-out effects.

2.4/ ATTACKS IN WSNS

As discussed before, WSNs suffer from limited computation capabilities, a small memory capacity, poor energy resources, absence of infrastructure, and susceptibility to physical capture. A variety of security solutions exists for infra-structureless networks (Ad hoc networks). Yet, they do not all answer the security challenges of WSNs.
WSNs are vulnerable to many attacks, due to their uncontrolled environment of deployment, the limitation of their resources, and the broadcast nature of the transmission medium. The attacks are mainly classified under two categories : physical attacks and non-physical attacks. In the following, we discuss some of the famous possible attacks in WSNs.

2.4.1/ NON-PHYSICAL ATTACKS

These attacks aim at disturbing the service and the normal operations in the network. In the following, examples of well-known non-physical attacks in WSNs are given.

2.4.1.1/ DENIAL OF SERVICE ATTACK

A Denial of Service (DoS) attack is an attempt to render the network unavailable to its users by leading to a server overload. It forces the network to either reboot or exhaust its resources, making it fail to perform the intended tasks. This attack is dangerous as WSNs lack the capacity to handle computational overhead to implement typical defensive strategies.
The DoS attack can target the physical layer (jamming, tampering), the link layer (collisions, exhaustion, unfairness), the network and routing layers (neglection, greed, homing, misdirection, black holes), or the transport layer (flooding, desynchronization) [Walters et al., 2007].
DoS attacks are very common. Therefore, there exist effective defensive measures against these attacks. Table 2.1 summarizes the possible defenses against different DoS attacks.

Kim *et al.* [Kim et al., 2006] proposed a DoS detection method reflecting the resource

Network Layer	Possible Attacks	Defenses
Physical Layer	- Jamming	- Spread-spectrum - Priority messages - Lower duty cycle - Region mapping - Mode change
	- Tampering	- Tamper-proof - Hiding
Link Layer	- Collision	- Error correcting codes
	- Exhaustion	- Rate limitation
	- Unfairness	- Small frames
Routing Layer	- Neglect and Greed	- Redundancy - Probing
	- Homing	- Encryption
	- Misdirection	- Egress filtering - Authorization - Monitoring
	- Black holes	- Authorization - Monitoring - Redundancy
Transport Layer	- Flooding - Desynchronization	- Client puzzles - Authentication

TABLE 2.1 – DoS attacks for different network layers [Wood et al., 2002]

constraints of sensors. Their approach relied on two types of entropy estimators. A main estimator is charged of the synthesis of localized computations, whereas the other estimators are deployed hierarchically according to the network topology.

Wood and Stankovic [Wood et al., 2002] described an approach to defend against jamming. In a first step, the nodes surrounding the jammed area report their status to they neighbors. In a second step, the neighbors collaborate to identify the jammed region so they will not route packets through it.

2.4.1.2/ SYBIL ATTACK

The Sybil attack was defined by [Douceur, 2002] as « an attack able to defeat the redundancy mechanisms of distributed data storage systems in peer-to-peer networks ».
A malicious device illegitimately takes on multiple identities to gain a large influence on the communication mechanism, by appearing and functioning as multiple distinct nodes. This attack is effective against routing algorithms, data aggregation, voting, fair resource allocation, and foiling misbehavior detection [Walters et al., 2007].
The Sybil attack is a harmful threat to sensor networks as it defeats redundancy mechanisms. To defend against such attacks, the network needs a mechanism to ensure that an identity is being held by one, and only one node in the network.
A light-weight identity certificate method is proposed by [Zhang et al., 2005]. This method avoids public key cryptography by using one-way key chains and Merkle Hash trees.

2.4.1.3/ TRAFFIC ANALYSIS ATTACK

Even with encrypted data, traffic analysis is able to determine what type of information is being communicated in the network (chat, requests...) and even falsify an identify and disable the base station [Walters et al., 2007]. A malicious node can generate a fake physical event to be sensed. The attacker, then, will watch the activity of the nodes in the network ; nodes tending to forward more data packets are closer to the base station.

Deng *et al.* investigate three techniques that aim at hiding the true location of the base station [Deng et al., 2004]. The first technique is multi-parent routing scheme. It introduces a degree of randomness in the multi-hop path from the source node until the base station. The second technique, called random walk, generates random fake routes to mislead an adversary while tracking the packet. And finally, fractal propagation consists in randomly creating multiple areas of high communication activities to disguise the true location of the base station.

2.4.1.4/ NODE REPLICATION ATTACK

In a node replication attack, an adversary can capture a sensor node, copy its ID, and insert a replica in the network [Parno et al., 2005].
The consequences of such an attack can be severe ; packets can be misrouted, changed, or corrupted, and significant parts of the network can be disconnected [Walters et al., 2007], etc.
[Parno et al., 2005] argue that previous node detection replication schemes depend either on centralized mechanisms or on neighboring voting protocols. Thus, they suffer either from single failure points or from failure to detect distributed replications. Considering these limitations, they propose two algorithms : randomized multicast and line-selected multicast. The line-selected multicast algorithm is inspired from the rumor routing described by Braginsky and Estrin in [Braginsky et al., 2002]. This algorithm helps reduce the communication cost of the randomized multicast protocol.

2.4.2/ PHYSICAL ATTACKS

In a WSN, adversary can perform the following physical attacks :

- Known-Plaintext Attack (KPA) : the attacker, having samples of both the plain text and the corresponding encryption, can reduce the security of the encryption key and reveal some of the information circulating in the network.
- Chosen-Plaintext Attack (CPA) : the attacker can choose a text to be encrypted. Doing so, it is easy to gain further information about the encryption key.
- Man-In-The-Middle Attack(MITMA) : the attacker creates a link between two nodes, through which they will communicate. The network cannot identify this connection as a malicious link. The attacker is then able to control the communication by intercepting and injecting messages.

2.5/ DEPENDABILITY OF A WSN

The dependability of a WSN is a property that integrates the attributes needed for the application to be justifiably trusted. Such a network should be able to deliver a correct service, i.e., a service that implements the system's function, and makes sure that a failed component will not lead to system failure. System dependability was defined by [Avizienis et al., 2000] as « the ability of a system to avoid failures that are more frequent or more severe, and outage durations that are longer, than is acceptable to the users ». In the following, we go through what the threatens the network's dependability, how to evaluate it, and how we can attain it.

2.5.1/ THREATS

Developing a dependable WSN starts with defining the dependability requirements of users. In order to satisfy these needs, it is crucial to understand what might stop the network from delivering a correct service. In this section, the threats that can affect the dependability of a WSN are enumerated.

2.5.1.1/ FAULTS

A fault is the cause of an error, and it indicates a defect in the system. Its presence does not systematically lead to a failure. A fault is considered to be active only when it produces an error of one or more components. It is considered as transient if it affects the communication links between the nodes, and permanent if it is caused by hardware malfunction [Silva et al., 2012].

In [de Souza et al., 2007], the authors classify the sources of faults under two main categories : node faults (related to hardware) and network faults (related to routing). A fault can have various origins, which are classified in Figure 2.6.

2.5.1.2/ ERRORS

An error takes place at run-time when some parts of the network enter in an unexpected (invalid) state that might result in a subsequent failure or undesirable outcomes. Such states are called hazards. As errors are hard to observe, special tools, such as debuggers, are required to declare their presence. An error indicates a discrepancy between actual behavior and intended behavior inside the network. It can then be detected if an error message or signal indicate its presence. If not detected, the error is called latent error [Avizienis et al., 2000, Taherkordi et al., 2006].

2.5.1.3/ FAILURES

A network failure is the observable consequence of an error. It occurs when the delivered service is no longer correct. The opposite transition (from incorrect to correct service) is called network restoration. Yet, the alteration of the service is not considered as a

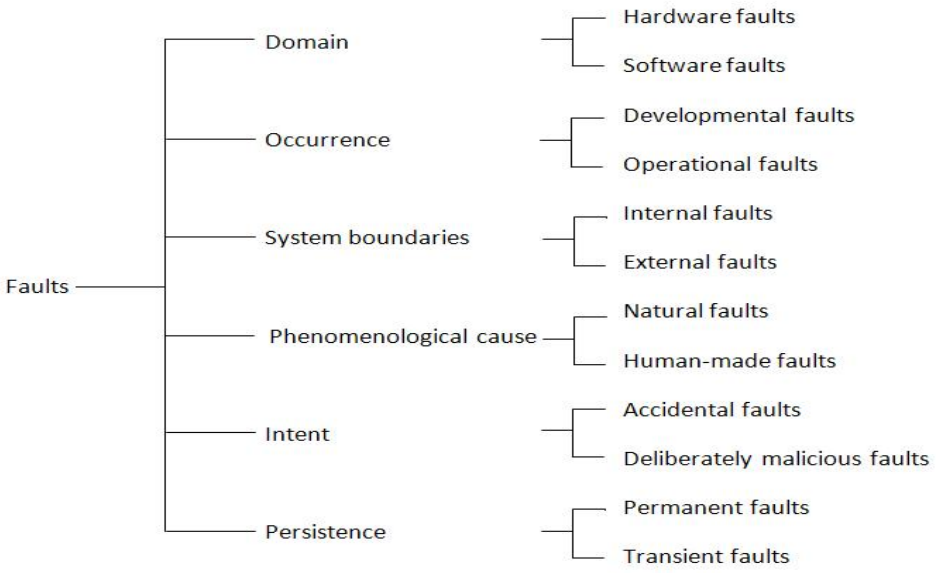


FIGURE 2.6 – Fault classes.

network failure until it reaches the service interface. With the implementation of fault tolerance techniques, a network failure can be avoided even when an error is activated [Avizienis et al., 2000, Silva et al., 2012]. The possible failure modes are outlined in Figure 2.7.

2.5.2/ ATTRIBUTES

This section deals with the ways by which we assess the dependability of a system. The attributes of dependability can vary in number and degree of importance considering the nature of the application and the intended service. The network, thus, is made dependable by adjusting the balance of the techniques to be employed according to the user's needs.

2.5.2.1/ AVAILABILITY

In the classical definition, a network is considered as highly available if its downtime is very limited. This can be due either to few failures, or to quick restarts when failures take place [Knight, 2004, Taherkordi et al., 2006]. If we add the security aspect, we can define availability as readiness for correct service for authorized users. This attribute can be computed as the probability that the network is functioning at a given time [Silva et al., 2012].

2.5.2.2/ RELIABILITY

A reliable network is a network that is able to continuously deliver a correct service. It can also be defined as the probability that a network functions properly and continuously in a time interval [Silva et al., 2012, Taherkordi et al., 2006].

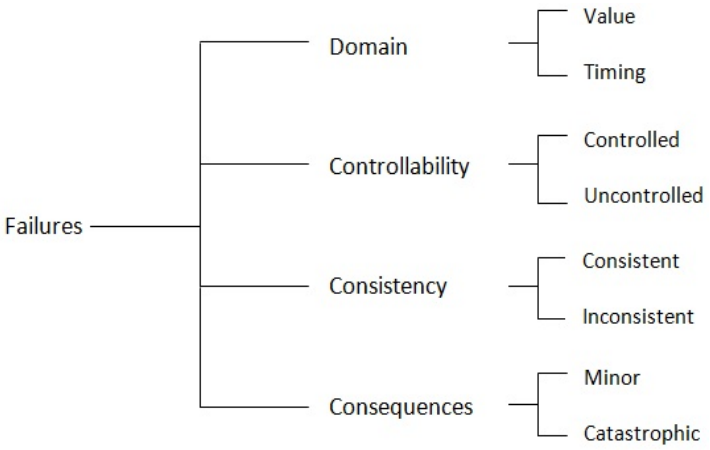


FIGURE 2.7 – Failure modes.

Most of the research works that have been accomplished so far employ retransmission mechanisms over redundancy schemes to achieve network reliability. The main purpose of a WSN is the correct delivery of data packets from sensor nodes to end users. Thus, reliability of WSNs is highly related to data transport. Reliability can be classified into different levels : packet reliability, event reliability, Hop-by-Hop reliability, and End-to-End reliability.
Both packet and event reliability levels deal with the required amount of information to notify the sink of the occurrence of an event within the network environment. Whereas the remaining two levels (i.e., Hop-by-Hop and End-to-End reliability levels) are concerned with the successful recovery of event information. Yet, all of them rely on retransmission and redundancy mechanisms.

Retransmission-based reliability : Packet retransmission is a very common technique to recover the loss of data packets which did not arrive to their destination. This is generally ensured through the use of acknowledgments. An acknowledgment is a signal (or a message) passed between two communicating terminals to validate the receipt of information, as a part of the implemented communication protocol.

There mainly exist three different acknowledgment mechanisms that a receiver can employ to notify the sender of the reception status. The reception of an explicit acknowledgment (eACK) is a guarantee for the sender that the message was successfully delivered. In the opposite case, negative acknowledgment (nACK) means that the packet did not arrive to its intended destination. These two mechanisms increase the transmission overhead and thus consume much energy. Implicit acknowledgment (iACK) reduces energy consumption and it requires the sender to take benefit from the broadcast mechanism by listening to the channel and interpret the reception of the packet.

To better explain this mechanism, an illustration of the eACK is given in Figure 2.8. When the receiver (Host B) receives a message from the sender (Host A), the receiver sends back an acknowledgment of receipt to the sender. Once Host A receives the acknowledgment, it can send the next packet. If ts acknowledgment is lost, the sender waits for a certain amount of time (called the timeout) which once elapsed, the packet is resent.

[Akan et al., 2005] presented the first event-based end-to-end reliability protocol. This algorithm, which is called Event-to-Sink Reliable Transport (ESRT), has the ability of self-

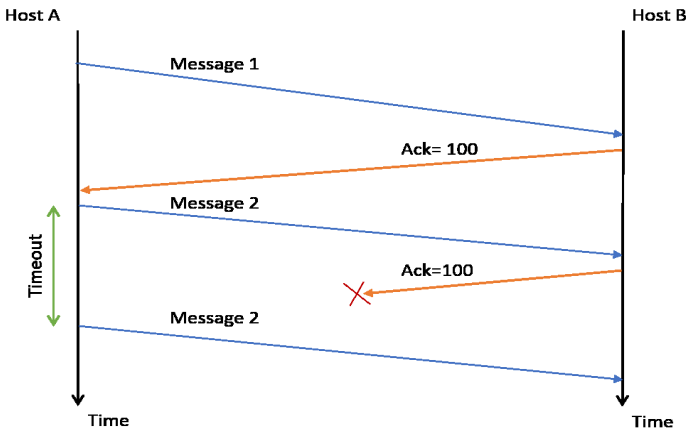


FIGURE 2.8 – Explicit acknowledgment mechanism.

configuring according to the network condition, thus it is robust in dynamic network topologies. Zhou *et al.* [Zhou et al., 2005] see reliability from a different angle compared to what is published in [Akan et al., 2005, Gungor et al., 2006, Iyer et al., 2005]. They consider that reliability cannot just be measured by the total of incoming packets at the sink node. It should, instead, refer to nodes contribution to improve the sink information about a certain phenomenon. Price-Oriented Reliable Transport (PORT) protocol [Zhou et al., 2005] also considers an in-network congestion-avoidance mechanism as a remedy to the drawbacks related to end-to-end schemes. In the proposed scheme, data packets avoid the paths with high loss rates. As a result, PORT is more energy efficient compared to ESRT [Akan et al., 2005] and DST [Gungor et al., 2006].

Redundancy-based reliability : Reliability can also be introduced via data redundancy mechanisms. A packet is transmitted in multiple copies using different routes as a backup plan in case one route fails (see Figure 2.9.

[Al-Wakeel et al., 2007] proposed a Path Redundancy Based Security algorithm (PRSA) that defines secure multiple least cost routing paths between source and destination nodes. The optimum paths are selected referring to Dijkstra algorithm described in [Dijkstra, 1974]. When a node is suspected of being malicious, it will be removed from the routing path. Furthermore, PRSA allows source node to transmit data packets using various modes in order to enhance network security.

[Mojoodi et al., 2011] studied the effect of redundancy on the number of correct responses of WSN on the received queries, and also investigate the change in the needed level of redundancy according to different network conditions. The simulations showed that redundancy is only efficient if the number of clusters needed to respond to the request is important or if error probability is high. In the opposite case, redundancy mechanisms will only lead to extra energy consumption and possible failures related to unnecessary communication cost [Mojoodi et al., 2011].

Redundancy mechanisms reduce the risks of losing information by increasing the chances of delivering the data packet to the intended destination. However, sending multiple copies exhausts both memory and energy resources as well as it increases the network latency.

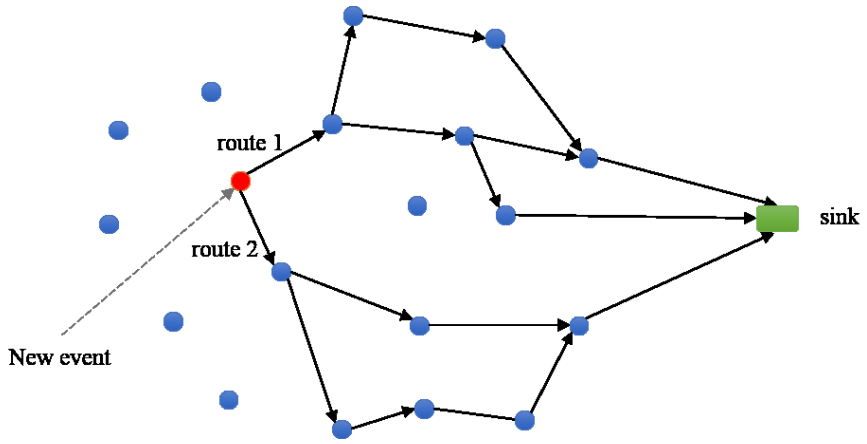


FIGURE 2.9 – Packet redundancy mechanism.

2.5.2.3/ SECURITY

WSNs are different from traditional computer networks. Therefore, existing security mechanisms are not suitable for these networks. Developing adequate security measures requires understanding WSNs' constraints that related to security issues.

Entity authentication : An attack on a network can be extended to more than just modifying the data packets originally circulating in the network. An attacker can inject additional data packets to disturb the normal function of the network and to tamper with the decision making process. For this reason, a receiver (i.e., node) must be sure that the data being accepted is coming from a member of the network. Similarly, a sender needs to verify that the reception entity is whom it claims to be. This finality can be achieved through authentication.
[Benenson et al., 2005] based their entity authentication on elliptic curve cryptography. Each user holds a legitimate certificate, which is the public key signed by a certification authority. Every node can verify the legitimacy of the users since the public key with the signature are preloaded in the sensors. Yet, this scheme requires an significant overhead for data encryption.

Backward and Forward Secrecy : By capturing a sensor node, or inserting a malicious one, an adversary tries to gain access to confidential information in the network. In order to prevent this from happening, a node joining the network must be forbidden from decrypting old messages (prior to it joining the network). Similarly, after node leaves the network, it must be forbidden from any further decryption.

Data confidentiality : One of the most important issues related to network security is data confidentiality, and it refers to limiting data access to legitimate destinations. Keeping data packets confidential mainly means that :

– Sensor readings can only be done by the legitimate destination ; a sensor node holding information must not leak information to its neighbors.

– Communication channel has to be secured, especially when the data being communicated is highly sensitive.
– The network needs to achieve confidentiality by encrypting data during transmissions.

[Bahi et al.,] argue that in-network communication, node scheduling, and data aggregation need to be proven secure. For this matter, they proposed a security framework for WSNs. The authors proved that in-network communication answers to security objectives (indistinguishability, non-malleability, detection resistance). In addition to this, the proposed algorithm is able to aggregate data over encrypted packets.

Data freshness : Data freshness means that data circulating in the network is recent and that no old messages are being replayed. In order to secure the network, shared keys need to be changed over time. During the update propagation time, an adversary can perform a replay attack, especially when the sensor node is unaware of key changing time.

Data integrity : Data packets need to be maintained safe and unchanged over their life-cycle. Even harsh environment can take part in altering data while being routed. This is why it is crucial to implement mechanisms ensuring that, for a data packet, information being sent is equal to the information being received.

Secure localization : Since there exists no physical connection between the nodes in a WSN, it is highly important for the network to be able to accurately locate each sensor in the network. Yet, to maintain the network's integrity, the locations need to be recorded as secret information.

Time synchronization : As discussed previously, energy is an issue for WSNs. For this reason, sensor nodes need to go to sleep when they are neither sending nor receiving a data packet. Thus, the network needs to ensure synchronization among the sensor nodes so those participating in the transfer process would be awake when they are needed.

2.5.3/ DEFENSIVE MEASURES

Key establishment techniques have received great attention for many years. Nevertheless, WSN applications are relatively recent. Besides, the features of these networks are different from traditional networks. Therefore, pre-existing techniques for key establishment are an unsuitable solution for WSNs applications. Traditionally, key exchange techniques use asymmetric cryptography (public key cryptography). Unfortunately, low powered WSNs are unable to handle such a computationally intensive technique.
The easiest way for encryption keys distribution, is to establish one single key for the entire network and forward it. It is easy to notice that this method is inefficient as one node can compromise the entire network.
An alternative solution that can be adopted is symmetric encryption key. This technique secures communication between two hosts as they share a private key that is not recognized by the rest of the network. This key will be used for both data encryption and

decryption.
Another possibility is random probabilistic key distribution scheme. The initialization stage starts with pre-loading in every sensor node a maximum number of keys (with respect to the memory). This is done in a way that two sets of keys (in two different nodes) will at least share one key. By broadcasting the identity of the keys, every node can discover the neighbors with which it can exchange information. Now, every node can only communicate with its legitimate neighbors ; a link only exists between nodes sharing a key. It is then possible for a sensor node to safely establish a link with a target node by secretly sharing a key via their neighbors [Li et al., 2011].

2.5.4/ MEANS TO REACH DEPENDABILITY

In this section we will discuss different ways to increase the dependability of a network.

2.5.4.1/ FAULT PREVENTION

Once detected, it is important to prevent the fault from being incorporated into the network. Fault prevention starts with the design of the network (efficient designing rules), through the implementation (simulations, structured programming), and during network operations (network maintenance, network protection).

2.5.4.2/ FAULT REMOVAL AND FORECASTING

Fault removal reduces the number and severity of faults within the network. It can be performed during the implementation of the network. The network design can be verified while being developed through verifications. Fault removal can also be performed while the network is being used. This is achieved through the maintenance cycle via corrective maintenance and preventive maintenance.
As for fault forecasting, it estimates the present number of faults, their future incidence and the likely frequencies of their future occurrences. It aims at removing the effects of faults before their occurrences.

2.5.4.3/ FAULT TOLERANCE

In order to prevent interrupting the network operation, it is important to carry out mechanisms that will allow the network to continue delivering the required service even in the presence of active faults. Fault tolerance is a technique that allows the network to continue delivering correct service until full recovery, without any interruption [Akyildiz et al., 2002]. [Geeta et al., 2013] presented a fault tolerant communication framework for WSNs regarding the remaining energy in sensor nodes.
Fault tolerance in WSNs is important for several reasons [Koushanfar et al., 2004] :

Technology and implementation aspects : a WSN is exposed to interactions with its environment, causing hardware degradation. Plus, the network is required to perform a variety of actions under energy constraints.

Complexity : the complexity of the application will grow as the complexity of architecture and technologies increases. This will render the testing phase more and more complicated.

Relatively recent scientific field : research related to WSNs is still an open field, where there is no best way to address a problem, and no mistake-free solutions.

2.6/ CONCLUSION

Wireless sensor networks are known for their easy deployment, their low cost, and their capability to extract localized features. Unfortunately, they have limited resources and modest computational capabilities. And due to the nature of communication, packet loss, data alteration, transfer delays, and some other issues are a constant threat to the network's reliability. The literature of dependability of wireless sensor networks reveals some tools that can be put in place, to improve the quality of service. Considering the limited memory space, the adopted solutions are preferably application-oriented. In other words, knowing our target application helps understand what could have a bad influence on the network performance. Thus, a solution can be deployed with a better use of the limited memory space.

In the next chapter, an overview on prognostics and health management is given. We will try to draw the challenges related to a wireless sensor network monitoring in order to propose solutions.

3

Prognostics and Health Management

Maintenance is an important activity in industry. It aims at either reviving a machine/component, or preventing it from breaking down. Different strategies have evolved through time, bringing maintenance to its current state (predictive maintenance). This evolution was due to the increasing demand of reliability in industry. Nowadays, plants are required to avoid shutdowns while offering safety, reliability, availability, and reduced costs [Peng et al., 2010].

In condition-based maintenance (CBM) and Predictive Maintenance (PM), monitoring devices are used to survey the physical condition of the equipment. When a certain level is reached (threshold), a maintenance activity should be performed to ensure the continuity of the machine/system's normal functioning. Comparing to corrective and preventive maintenance, CBM and PM require extra investments related to the monitoring equipment. Nevertheless, it increases the system's availability and optimizes the service life, all while reducing downtime and avoiding unnecessary maintenance.

In order to be efficient, a CBM program needs to go through the following steps [Jardine et al., 2006], as illustrated in Figure 3.1.

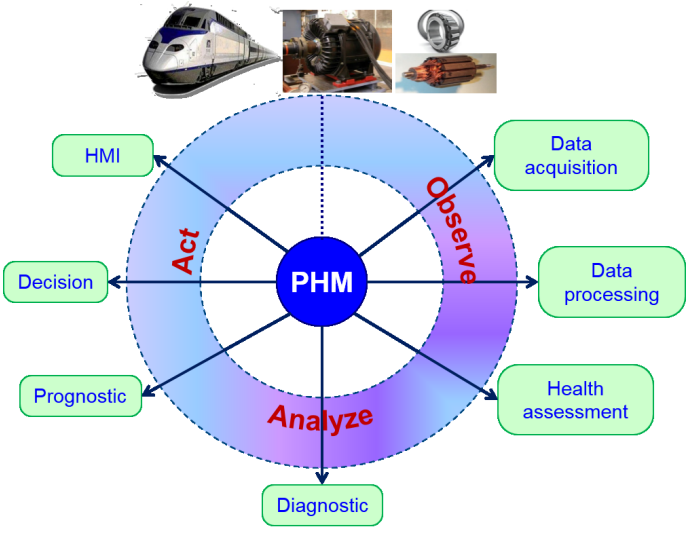


Figure 3.1 – CBM Flowchart.

In this chapter, CBM and PM will be reviewed in details. We will go through the reported

strategies in the literature and discuss their advantages and drawbacks.

3.1/ CBM IN DETAILS

The use of condition-based maintenance is privileged due to its advantages comparing to other strategies. A comparison is summarized in Table 3.1.

Strategy	Advantages	Drawbacks
Corrective maintenance	- Low cost - Service life is fully exploited	- Sudden failures - High repair costs - Dangerous failure outcomes
Preventive Maintenance	- Improved availability - Decreased downtime - Costs saving	- Unplanned downtime - Unneeded repair costs - Interrupted service life
Condition-based and predictive maintenance	- Increased availability - Reduced downtime - Reduced costs - Optimal system service	- Investment in monitoring equipment

TABLE 3.1 – Comparison of maintenance strategies.

PHM is the core activity of CBM and PM, and it implies the same steps. In the following, we will briefly discuss these steps, namely : data acquisition, data processing, health assessment, diagnostics, prognostics, and decision making support.

3.1.1/ DATA ACQUISITION

Prognostics and Health Management (PHM) requires information about the targeted physical assets. The data is acquired by means of sensors which are placed on/around the critical components. The stored data will later on be fed as inputs to the developed algorithm for health assessment, diagnostics, and/or prognostics. We can see here the importance of the quality of the data as all of the next steps rely on these measurements. The gathered information can contain either event-data or Condition Monitoring (CM) data.
Event-data reveal what happened, what were the causes, and what was done (repair, breakdown, installation, etc.). On the other hand, CM data contain measurements are related to the machine's condition (pressure measurements, environment data, etc.). These two types of information are equally important for health assessment, diagnostics, ans prognostics.
In Figure 3.2, a summary of data acquisition steps is given.

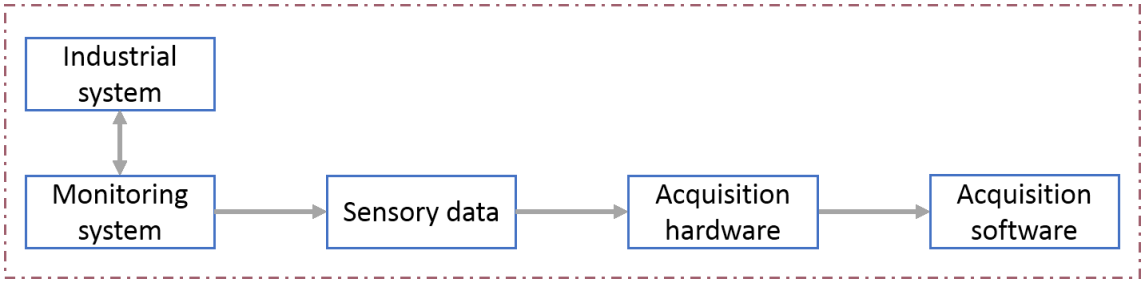


FIGURE 3.2 – Data acquisition system.

3.1.2/ DATA PROCESSING

Once the information is available, and before building the degradation model, it is very important to perform data cleaning in order to enhance the results of health assessment, diagnostics, and prognostics. Due to the errors that might be induced by the communicating channel, the importance of processing the data consists in providing signals that are robust against the variations that might affect the raw data [Russell et al., 2003]. It aims at isolating all the possible faults and avoiding the so-called "garbage in, garbage out" problem. Data processing was defined by [Bae et al., 2014] as the collection and manipulation of items of data to produce meaningful raw data.

In practice, raw sensor signals can be complex, and data describing the degradation is not easy to read. Reported data can have a value type, a waveform type, or a multi-dimensional type. The two last types can contain noise and thus be very hard to exploit. Data processing is an important step as it converts raw data into useful information. Many processing techniques have been reported in the prognostic literature [Tobon-Mejia et al., 2012a, Niu et al., 2010], like wavelet decomposition, data denoising, data smoothing, etc.

Data processing can be divided into two main tasks : (i) pre-processing of sensor raw signals and (ii) data analysis for more information extraction. This step aims essentially at improving the signals received from the monitoring device, providing a better understanding of the process that generated the data, enhancing the degradation model, and rendering the computation more effective by reducing the measurement size. The steps of a data processing system are given in Figure 3.3.

3.1.3/ HEALTH ASSESSMENT

Health assessment consists in determining the system's state of health at a given time. To do so, sensory data are reported periodically to monitor critical components. These data correspond to measurements of relevant parameters (pressure, temperature, moisture...), and are useful to assess the machine's condition. Thresholds related to the monitored parameters are fixed. Once a threshold is reached, the system is considered to be in the corresponding state (see Figure 3.4).

Health assessment task can be narrowed down to a classification problem, and there exist several ways to perform this task. For instance, human expertise can be the basis for identifying the class to which belongs a functioning profile. This necessitates a solid

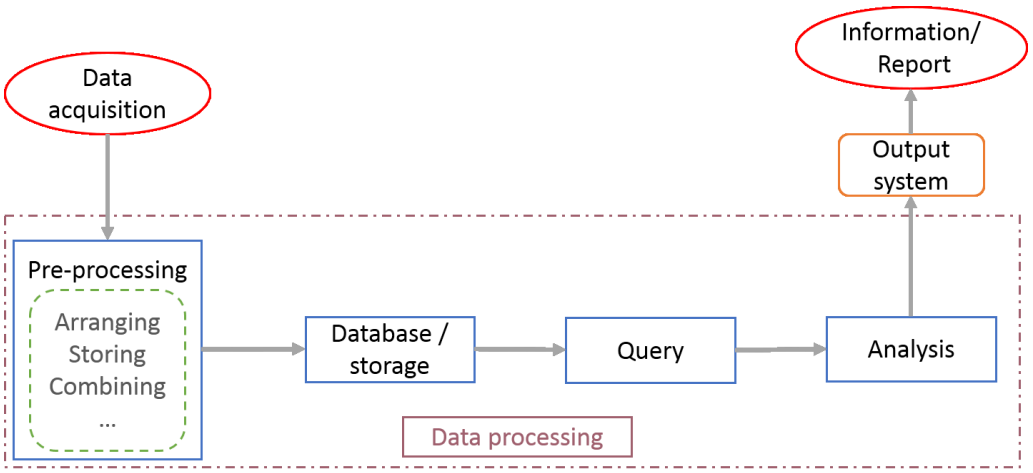


FIGURE 3.3 – Data processing system.

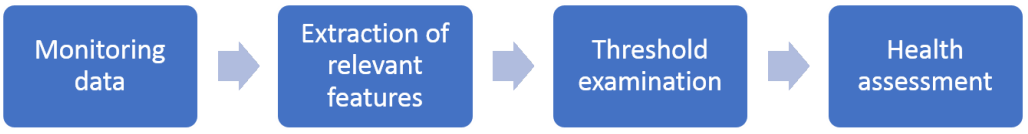


FIGURE 3.4 – Health assessment process.

knowledge of the application domain based on both experience and history of observations. This knowledge will help infer rules relating an observation to its meaning. This knowledge can also be explored to develop mathematical models describing the physics of the system under consideration. On-line measurements can be run through the developed model to identify the health state. Machine learning techniques can also solve this problem. Off-line observations will be used to train a model which will then be used on-line to determine the class of new observations.

3.1.4/ DIAGNOSTICS

Diagnostics is performed after the fault takes place. It identifies the fault's type, size, location, and cause. Diagnostics aims at relating the cause to the effect. It is an understanding of the relationship between what we observe and what happened before [Sikorska et al., 2011].

In Figure 3.5, the successive steps of a diagnostic process are illustrated.

FIGURE 3.5 – Diagnostic's different steps.

– Fault detection is the process of reporting an anomaly in the system's behavior.
– Fault isolation is charged of determining and locating the cause (or source) of the problem. It identifies exactly which component is responsible of the failure.

– Fault identification aims at determining the current failure mode and how fast it can spread.

3.1.5/ PROGNOSTICS

While diagnostics aims at identifying and quantifying an actual failure, prognostics has the goal of anticipating failures. Several definitions concerning prognostics exist in the literature. We summarized some of them in Table 3.2.

Definition	Authors	Reference
Estimation of time to failure and risk for one or more existing and future failure modes.	ISO 13381-1	[ISO13381-1, 2004]
Estimation of the time before failure, or the remaining useful life, and the associated confidence value.	Tobon-Mejia *et al.*	[Tobon-Mejia et al., 2012a] [Tobon-Mejia et al., 2012b]
Indicates whether the structure, system, or component of interest can perform its function throughout its lifetime with reasonable assurance and, in case it cannot, to estimate the remaining useful life.	Zio and Di Maio	[Zio et al., 2010]
Predicts how much time is left before a failure (or more) occurs, given the current machine condition and past operation profile.	K.S. Jardine *et al.*	[Jardine et al., 2006]

TABLE 3.2 – Some definitions of prognostics reported in the literature.

Prognostics considers past events, the machine's current state, and operating conditions to estimate the RUL. This estimation is done by inspecting the evolution of continuous measurements of parameters that need to be monitored in time to assess the machine's state. These parameters can be temperature, humidity, vibration, pressure, and so on. A monitored parameter has a fixed threshold. Once reached, an alarm goes off indicating that a symptom of system deteriorating has been detected. The RUL is then computed with an associated confidence limit. The latter information illustrates to what point the predictions are trustworthy. The uncertainties of the RUL predictions have two causes : either the threshold value of monitored parameter, or the RUL prediction itself.
In Figure 3.6, we can observe the uncertainties that can be related to RUL prediction.

An efficient prognostic activity requires the collection of documented data that covers the machinery and components under consideration. All monitored parameters and descriptors need to be available with historical records of operations and events. Failure identification and initial diagnostics are also mandatory to improve the prognostics results [ISO13381-1, 2004].

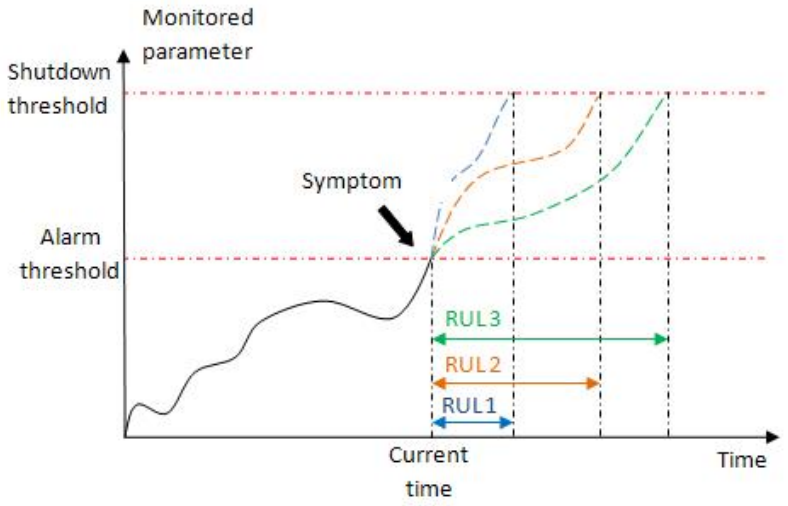


FIGURE 3.6 – An illustration of RUL with uncertainties.

3.1.6/ DECISION SUPPORT SYSTEM

Once prognostics are performed and RUL is estimated, the next step is to decide what are the actions that need to be taken (repair, replacement, maintenance, oil changing...). Decision making is a cognitive process. It consists of selecting an action among different possible scenarios, to produce a final choice. The general process is described in Figure 3.7.

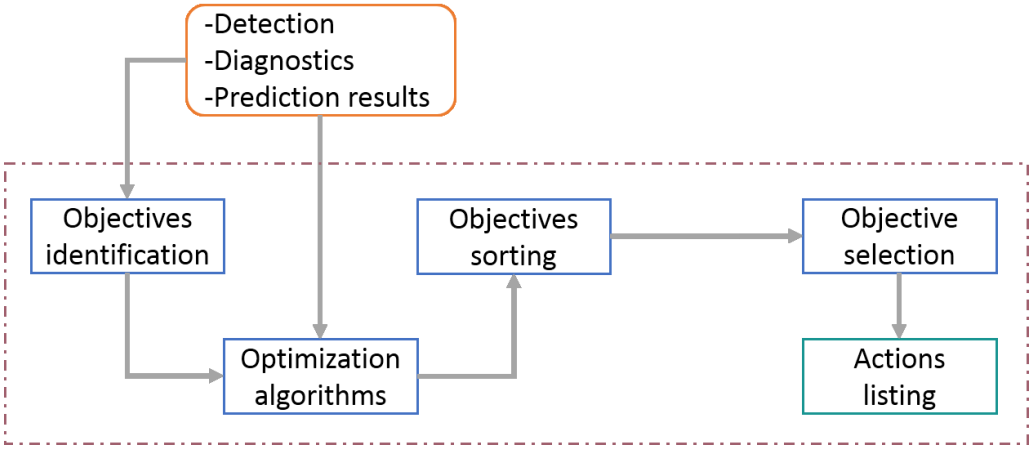


FIGURE 3.7 – Decision support system.

First of all, the objectives need to be established. An objective can be keeping component from failure until next inspection, reducing overall costs, or any other purpose a plant can be aimed for. All the objectives are then classified in order of priority and importance. Alternative actions are developed to answer the established objectives, and the actions that are able to satisfy most of the objectives are selected.
A decision needs to be made to select the appropriate action. This can be done by implementing a tool among different possibilities.

– **Domain experts :** it is very often that plants trust the advices provided by engineers

and domain experts. Thanks to their knowledge and experience, they are able to point out good solutions and uncover the limitations related to a strategy.
– **Eliminations :** another solution is to eliminate non-realistic solutions one by one, or compare them in a pairwise manner. At the end, the remaining option is selected.
– **Analytic networks :** these networks provide a hierarchy of the selected action with goals, alternatives, and consequences.
– **Simulations :** there are many graphical tools used to visualize the behavior of a system under different conditions. Simulations are a popular tool for decision making support as they offer clarity and possibility to alter criteria while simulating.

3.1.6.1/ HUMAN MACHINE INTERFACE

The human machine interface (HMI) is the user interface in a manufacturing or process control system. It is the space where the interactions between humans and machines take place. The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operators' decision making process. HMI provides a graphic-based visualization of an industrial control and monitoring system. It is a software application that presents information to an operator or user about the state of a process, and to accept and implement the operators control instructions. Typically, information is displayed in a graphic format (Graphical User Interface) to illustrate the analysis results in a comprehensive way. Figure 3.8 gives an illustration of this interaction.

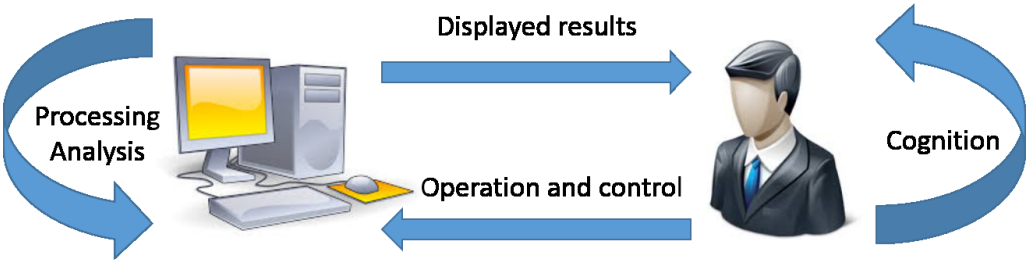


FIGURE 3.8 – Human machine interaction.

3.2/ CLASSIFYING APPROACHES

Prognostics approaches are classified under groups employing, more or less, the same techniques. Nevertheless, researchers use different classifications. Table 3.3 summarizes the different groups we encountered during our study More details on each approach can be found in the given references.

In this thesis, we consider four groups : Physical models, Knowledge-based models, Data-driven models, and Hybrid models. They are illustrated in Figure 3.9 and detailed in the following sections.

Classifying groups	Authors	Reference
- Statistical approaches - AI approaches - Model-based approaches	K.S. Jardine *et al.*	[Jardine et al., 2006]
- Event-based prediction - Condition-based prediction - Integrated approaches	Heng *et al.*	[Heng et al., 2009]
- Physical model-based methodology - Knowledge-based methodology - Data-driven methodology - Combination model	Peng *et al.*	[Peng et al., 2010]
- Knowledge-based models - Life expectancy models - Artificial neural networks - Physical models	Sikorska *et al.*	[Sikorska et al., 2011]
- Model-based techniques - Model-free methods	Cadini and Avram Zio and Di Maio	[Cadini et al., 2009] [Zio et al., 2010]
- Model-based approaches - Data-driven approaches - Hybrid approaches	Hu *et al.*	[Hu et al., 2012]
- Model-based prognostics - Data-driven prognostics - Experience-based prognostics	Tobon-Mejia *et al.*	[Tobon-Mejia et al., 2012a]

TABLE 3.3 – Classifying models in the literature.

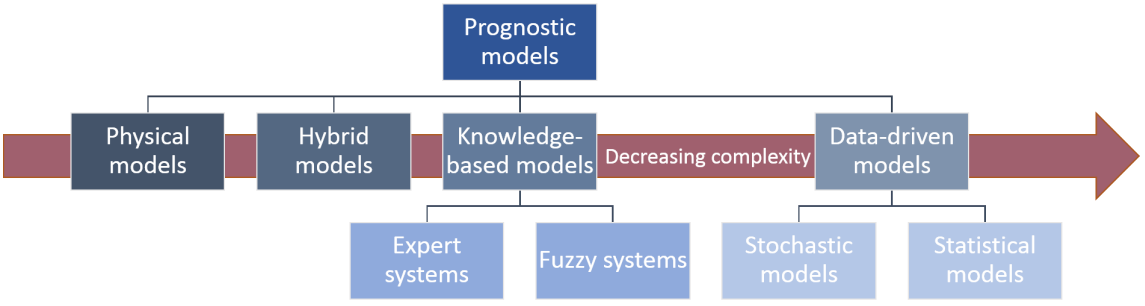


FIGURE 3.9 – Prognostic approaches.

3.2.1/ PHYSICAL MODELS

Physical models rely on mathematical models to describe the physics of a failure and are developed by domain experts. For this reason, the first condition for a reliable model is a good understanding of the behavior of the system responding to stress. The description of the behavioral models is realized via differential equations, state-space methods, or simulations.

In Figure 3.10, the general flowchart of a model-based approach is given.

Physical models are considered if :

– the mathematical model of the system is known ;

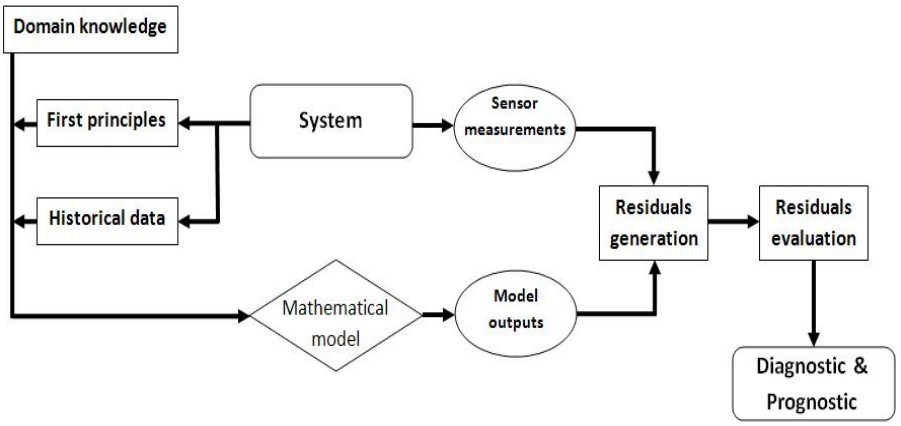


FIGURE 3.10 – Flowchart of a model-based approach.

– the failure mode is well understood ;
– a physical model for each failure mode is available ;
– the operating conditions can be monitored ; and
– data describing the conditions related to each process is available.

3.2.2/ KNOWLEDGE-BASED MODELS

Since it is really hard to build an accurate physical model for complex industrial systems, the employment of the latter is really limited. Besides, it is impossible to apply a developed model to a different component. Other methods, such as knowledge-based ones, appear to be promising as they require no physical model.
In the following, two examples of this approach are presented.

3.2.2.1/ EXPERT SYSTEMS

Since late 1960s, expert systems seemed to be suitable for problems usually solved by human specialists. These models consist of computer system, designed to display expert knowledge. This knowledge is extracted by domain specialists and organized into rules learned by the computer to generate solutions.
The general process of building such a model is described in Figure 3.11.

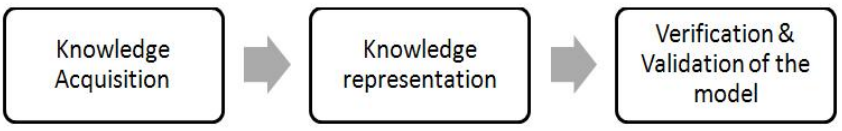


FIGURE 3.11 – Process of building an expert model.

Knowledge is usually acquired through human observations, sensor measurements, historical records, etc. Domain experts use all the knowledge they can gather and represent through comprehensive rules that relate an observation (or a situation) to its meaning.

The built model will next be tested in order to be validated and later used for on-line assessments.

The developed rules have the form of :

IF condition, **THEN** consequence.

Such a rule is strict and does not adapt to any changes in operating conditions. The only way to adapt the model to new situations is to add new rules whenever a new condition is observed. This can lead to a combinatorial explosion, especially that a rule is required for every possible combination of inputs. Another limitation of this model is that it is only as good as its developers.

3.2.2.2/ FUZZY LOGIC

Is a form of probabilistic knowledge, where the rules are approximate rather than fixed and exact. It was introduced by Lotfi A. Zadeh in 1965 [Zadeh, 1965].
The difference between fuzzy logic and classical predicate logic, is the use of fuzzy sets rather than discrete values standing for true or false. In a fuzzy set, variable's membership is defined based on their degree of truth. The truth value ranges from 0 (completely wrong) to 1 (completely true).The rules may look like :

IF condition « A » **AND** condition « B » **THEN** consequence.

The description associated to the parameters differs from the description used with expert system rules. Here is an example to illustrate the difference :

Expert system : **IF** engine is hot **THEN** shutdown.

Fuzzy logic : **IF** engine is slightly hot **AND** temperature is rising **THEN** cool down the system.

This new way of introducing rules gives the computer a very human-like and intuitive way of reasoning with incomplete, noisy, and inaccurate information. As a result, fault detection and prediction are more accurate, and for this reason, fuzzy logic is usually incorporated with other techniques.
Even though this method can only be developed by domain experts, it is easy to understand the developed rules. It is not only recommended because it covers a large set of operating conditions, but also because of its efficiency when it is impossible to build a mathematical model or when data contains high levels of uncertainties and noise.

3.2.3/ DATA-DRIVEN MODELS

In data-driven approaches, models are directly derived from condition monitoring data, based on statistical and learning techniques. These models have a double role : assess current operating conditions and predict the RUL. Neither human expertise nor comprehensive system physics are needed for the prognostic model building process.
A data-driven prognostic model transforms raw data provided by the monitoring system

into useful information. By the means of this information and historical records, a behavioral model is built and predictions can be performed. The building process is detailed in Figure 3.12.

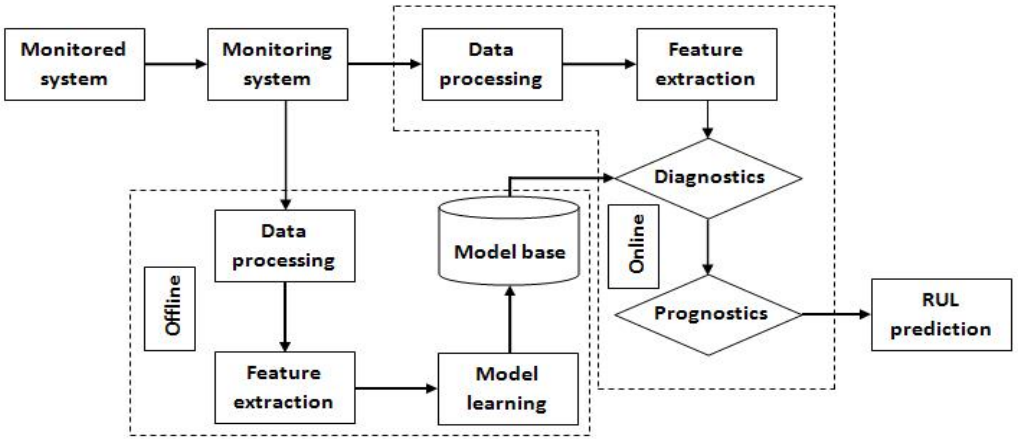


FIGURE 3.12 – General process of a data-driven approach.

The data-driven approach is popular and widely used because it offers a tradeoff between complexity and precision. This approach remains the best solution when obtaining reliable sensor data is much easier than constructing mathematical behavioral models. Nevertheless, accuracy depends on many factors.

– The training set : normally, an efficient training requires a large set of inputs. It is not easy to decide whether the amount of inputs we dispose is enough for training a reliable model or not.
– Operating conditions : manufacturing conditions change all the time, so do the environmental and operational conditions. All these changes may lead to uncertainties in the predictions as they refer to new situations that may not be recognizable by the model.
– Sensory signals : the amount of effective sensory data available when prediction is performed has an impact on accuracy.
– Degradation trend : RUL prediction relies on historical data and past events. As shown in Figure 3.6, the prediction is an extrapolation of what we observe up to the present moment. If the degradation trend is highly similar to a trend the model can recognize, prediction can be accurate (and the other way around).

In the following, we present some data-driven approaches.

3.2.3.1/ AGGREGATE RELIABILITY FUNCTION

For over 30 years, aggregate reliability function has been one of the most tools for prognostics used by industry. This method consists of accumulating information about failures history of a population, analyzing these trades, and finally defining a probability density function for the extracted hazard function. The outputs are indicators of when the failures are expected to happen. [Crevecoeur, 1993, Duane, 1964, Goode et al., 2000, Lee, 1980, J.M.Noortwijk, 2009, Todinov, 2005]

3.2.3.2/ ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks (ANN) mimic the human brain structure. Thanks to their complex layer structure, they are able to process complex non-linear functions with numerous inputs and outputs. Their popularity is due to their ability of providing a mathematical representation of the failure process, derived from from monitoring data instead of physical understanding of the deteriorating process. Besides the learning in neural networks is fast, and they were proven to be at least as good as the best traditional statistical model. A typical ANN, as shown in Figure 3.13, contains a layer of input nodes, one or more layers of hidden nodes, a layer for output nodes, and weights connecting perceptrons of different layers. These weights get adjusted through observations during the training process.

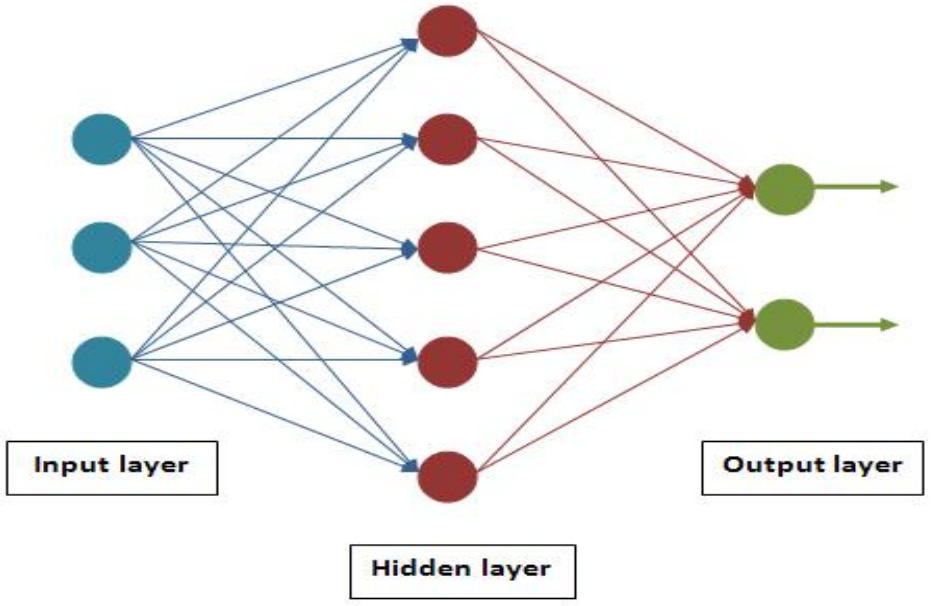


FIGURE 3.13 – Feed-forward artificial neural network.

ANN training can be either supervised or unsupervised. In a supervised learning, the network needs external inputs, like experimental data with known results, forming *a priori* knowledge about outputs. Feed-Forward (FFNN) is one of the most widely used neural network in supervised learning for diagnostics and pattern recognition. The current outputs of this model are completely independent of previous ones, and this can make the training process slow to converge. It is also hard to define the initial structure of the network and the optimal numbers of nodes and hidden layers. In the other hand, the Cascade Correlation Neural Network (CCNN) does not require neither initial structure for the network nor a predefined number of nodes. Instead of adjusting the nodes in a fixed topology, CCNN begins the training with a minimal network and adds perceptrons one by one. Once a hidden layer is complete, the corresponding input weights become fixed.

This fastens learning process reduces the complexity, and optimizes the network topology. In unsupervised learning technique, the network learns with the information it disposes and thus no external inputs are needed. Self Organization Maps (SOM) are able to produce low-dimensional outputs of high-dimensional data. They are different from other neural networks as they use neighborhood functions to keep the to-

pological properties of the input space. Recursive Neural Network (RNN) stores outputs and feeds as time-delayed inputs in order to help the weights adjustment and the network converge faster [Huang et al., 2007, Herzog et al., 2009, Wang et al., 2004, Brotherton, 2000, Tsui et al., 1995]. The major limitations and challenges to face while training an ANN are :

– A good training needs a large data set ; such a set is not easy to acquire and makes the training process time consuming. Besides it is a hard task to define the adequate amount of data for the training.
– Defining the number of hidden layers, as well as the activation function and nodes number for each layer can be a tough job.
– ANN lack of transparency, and the model accuracy needs to be balanced against the adaptability to unseen data.
– ANN lack of transparency, and the model accuracy needs to be balanced against the adaptability to unseen data.

3.2.3.3/ AUTO-REGRESSIVE MOVING AVERAGE

$ARMA$ models describe a weakly stationary stochastic process. The general model was described by Peter WHITTLE in 1951. These models are used for stationary data as they remove temporal trends. The model building process goes through three steps : model identification, parameter estimation and model validation. These steps are repeated until a satisfactory model is obtained.
$ARMA$ models are only effective for short term predictions. This is due to dynamic noise, high sensitivity to initial conditions, and errors accumulation in prediction process. [Wu et al., 2007, Yan et al., 2004]

3.2.3.4/ BAYESIAN TECHNIQUE

Bayesian networks are direct acyclic graphs that are a synthesis of probability and graph theory illustrating random variables and probabilistic inter-dependencies. They are a set of nodes representing different states, and directed edges describing the transition probability between those states. This illustrated in Figure 3.14.

Bayesian networks can be either static or dynamic. In both cases, their implementation requires modeling experts and know causes of route failures ; it is impossible for a Bayesian technique to predict a new failure mode that has not been observed before.
In a Dynamic Bayesian Network (DBN), it is possible to perform updates on the network. Therefore further states can be predicted. It is also assumed that :

– The network has the same structure at any time slice.
– The value of predictions at a time slice depend only on the previous one.
– The set of variables and probability definitions are the same for each time slice.

The most commonly used DBNs variants are Markov models, Kalman filters and Particle filters [Cadini et al., 2009, Kallen et al., 2005, Weidl et al., 2005, Weidl et al., 2003, K.Ito et al., 2000, Haug, 2005, Marquez et al., 2005].

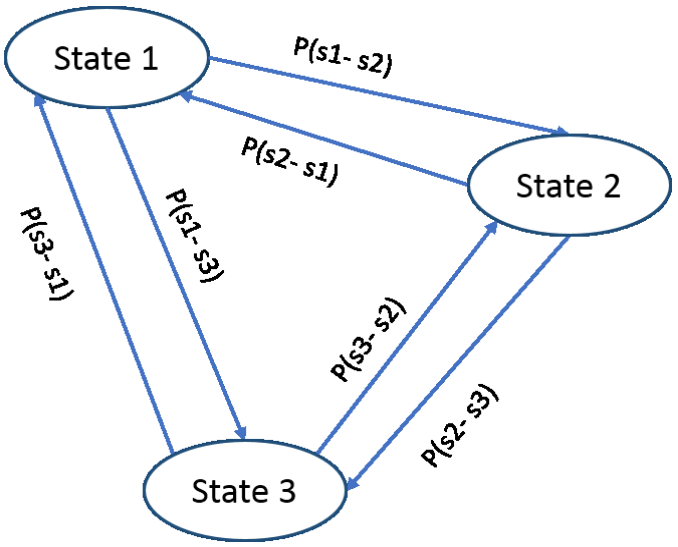


FIGURE 3.14 – An example of a Bayesian network.

3.2.3.5/ HIDDEN MARKOV AND SEMI-MARKOV MODELS

Markov models are the simplest of DBNs. They assume that the system, at a given time, can only be in one state. They are able to estimate the future by defining a probability of every possible transition in the network.
These models have their own limitations. (1) They depend only on most recent past states, so when they are applied to predict the future, they just consider the last state. This is not practical for real machines. (2) The probability of going from a state to another does not change in time ; a change in operating conditions cannot be considered. (3) A state can be either failed or not failed.
Semi-Markov models are useful for RUL prediction as the sum of probabilities of one state can be less than 1. When it comes to prognostics, Hidden-Markov Models (HMM) and Hidden Semi-Markov Models (HSMM) are used rather than classical Markov Models. In fact, for a running machine, not all the states are directly observable. HMM and HSMM perform fault classification by observing an output and linking it to an unknown machine condition. The outputs are dependent on non-visible states [Bunks et al., 2000, Baruah et al., 2005, Medjaher et al., 2012, Dong et al., 2007a, Dong et al., 2007b].

3.2.3.6/ KALMAN FILTERS

A Kalman Filter is a recursive processing technique for dynamic systems. It estimates the state with minimized mean squared error. The model building process consists of two phases : a prediction phase and a correction phase. Using prior knowledge of the state and outputs of previous estimations, the model provides estimations of current state with an associated error covariance. The state prediction is then compared to actual observation. Thus, the predicted error is adjusted and fed, with estimation value, as inputs for next prediction.

3.2.3.7/ PARTICLE FILTERS

Also called Monte Carlo method, Particle Filters are an estimation technique based on simulations. While Kalman filters perform predictions by extrapolating previous observations in the future, Particle filters use probability density function to select efficient samples for prediction. These samples are inputs to simulate the system's behavior and predict the state. With an appropriate sample size, Particle filters are more accurate than Kalman filters. Nevertheless, like all simulators, when the number of iterations increases, the model is more likely to degenerate.

3.2.3.8/ TREND EXTRAPOLATION

Trend extrapolation is a technique that analyzes a monitored parameter's trend. The analysis considers a single parameter plotted as a function of time. The observed trend is extended beyond the known data regions to determine the likelihood of the system's behavior. An alarm of end of life corresponds to the value of the monitored parameter, that once reached, the system would stop functioning. An estimation of RUL consists of extrapolating current trajectory in the future. The time at which the predicted trajectory reaches the threshold corresponds to the estimated RUL. It is possible to set more than one alarm (for maintenance actions), yet these alarms need to be carefully defined ; otherwise, components might be replaced when there is no need to, or the system will shut-down before any actions can be taken [Batko, 1984, Kazmierczak, 1983, C.Cempel, 1987].

3.2.4/ HYBRID MODELS

Usually, prognostic activity does not consider one parameter. The monitored parameters are diversified, as a consequence, it may be impossible to study failure behavior using only one model.

Hybrid models aim at improving prediction quality by providing more accurate RUL. All research works agree that physical models guarantee the most precise prediction. Nevertheless, even with good outputs quality, the complexity is too important to ignore. This complexity can be reduced by adopting a data-driven approach. Thus, we can take benefits from the merits of both prognostic approaches.

When physical understanding of failure mechanism and monitoring data are available, a hybrid approach is the best solution offering a compromise between model complexity and prediction accuracy.

The flowchart of a hybrid model is illustrated in Figure 3.15.

Table 3.4 is a summary of each model's advantages and drawbacks.

Prognostic Model	Advantages	Drawbacks
Physical models	- Require little data for prediction - Accurate and precise estimations - Can be reused in different conditions	- Very complex to build - Generates overall estimates - Needs complete knowledge of system behavior - Component and defect specific - Model validation requires a large set of data

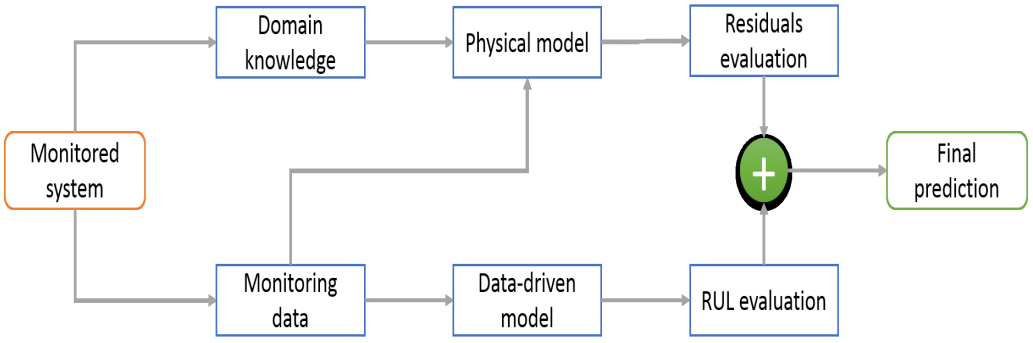


FIGURE 3.15 – General process of a hybrid approach.

Knowledge-based models Expert system	- Easy to develop and understand - Mimic human thinking	- Hard to acquire domain knowledge - Accuracy requires many rules - Hard to convert knowledge into rules - Does not generate a confidence limit, nor accurate RUL
Fuzzy logic	- Handles imprecise and incomplete data	- Rules need to be developed by domain experts - Decisions not easy to make
Data-driven models Aggregate reliability functions	- Simple and easy to understand	- Failures have to be statistically independent and equally distributed - No warnings before failure
ANN	- Handles online pattern recognition - Does not require any physical understanding of system behavior - Models complex systems	- Only efficient for a small set of data - Needs retraining with every change of conditions - Needs pre-processing in order to reduce inputs
ARMA models	- Efficient for real-time applications - Does not need data related to failure history - Does not require a detailed understanding of failure mechanism - Accurate short term predictions	- Significant data for training - Does not benefit from prior knowledge
Bayesian technique	- Does not require event data - Predictions are easy to establish	- Relies on accurate thresholds - Needs a lot of state transitions for efficient prediction

	- Handles incomplete data	- Unable to predict unanticipated failures - Prior knowledge needs to be available
HMM and HSMM	- Recognizes different failures - Able to recognize the change process - Models spatial and temporal data - Failure trend does not need to be monotonic - Manages incomplete data sets	- Intensive computations - The assumptions are not always practical - Large amount of data for training - Unable to predict unanticipated failures - State/Failure mode relation not clear
Kalman Filters	- Models multivariate and dynamic processes - Handles incomplete and noisy data	- Can diverge easily
Particle Filters	- Provides non-linear projections	- Requires a large number of samples - Not efficient for multi-dimensional data
PDF	- Accurate prediction close to failures - No need for CM data	- Requires an important sample size
PHM	- Simple to develop	- Includes strict assumptions - Requires historical data
Trend extrapolation	- Easy to apply	- Relies totally on past events - Needs a well-defined monotonic failure trend
Hybrid models	- Can be used with lack of historical data - Accurate predictions	- Requires both event and condition data

TABLE 3.4 – An overview of Prognostic models.

Model-based and hybrid approaches can be used when the physics of the failure mode is comprehensive. However, if the monitored system is complex, developing an accurate model becomes a very challenging task and the verifications become very expensive. The industrial revolution has led to a growing complexity of the engineering assets. Understanding the interactions between the different components failure modes and modeling all that by a mathematical model is almost impossible.
The increasing number of system components and the complexity of their interactions also results in a high number of rules describing the failure mechanism. A domain expert will spend an important amount of time trying to cover all the possibilities, and even if the task is successfully achieved, system validation and computations will be very expensive time-consuming.
Therefore, data-driven models are continuously gaining popularity in industry research due to their fast training and easy development. Comparing to the other approaches, a data-driven model has the particularity of being flexible (rather than component-oriented) and easy to reuse.

In this thesis, we omplement a data-driven approach for industrial health assessment. More precisely, we focus our attention on ensemble methods and propose the Random Forest (RF) algorithm to assess the machine's health with on-line monitoring. The deployment of this approach is also motivated by the research advances in data handling

(storage and processing) and sensor systems and networks, which opens new horizons in both industrial research and applications. We consider, in our work, the case where the system is monitores with a WSN. The choice of this network is further explained in the next section.

3.3/ WIRELESS SENSOR NETWORKS FOR INDUSTRIAL PHM

Reliability has become very essential in industry. It is a means to economic gain in addition to client trust. The research in the prognostics over the past years has resulted in a variety of tools and techniques offering plants the possibility to survey their systems, anticipate failures, and schedule maintenance activities. As the existent tools are different from one to another, they have different advantages, drawbacks, complexities, etc. Data-driven prognostic models drew a great deal of attention due to their low cost, low complexity, and easy deployment. The prediction model will first acquire information about the monitored system, assess the current state, and then extrapolate the health state in the future.

Data gathering is usually based on the deployment of independent sensors. This choice is not always feasible. A complex system, or one that spreads on a large area would require more than one parameter to be monitored, and therefore as many sensors for the monitoring activity. Connecting each of these sensors separately to the base station may render the wiring a complicated process. As consequence, connecting the sensors in a network is privileged. The use of wired networks gives rise to a number of drawbacks. Physical wires come with a price that grows proportionally to the their length and weight. For example, on an Airbus A380, we can measure 500 miles of cable weighing over 3 tons [1]. The use of WSNs would considerably decrease the weight, consequently fuel consumption, and therefore reduce the costs. Generally speaking, a WSN renders monitoring zones more accessible, by eliminating the difficulties related to wiring the network in-site. In addition to reducing the complexities of manufacturing, WSN reduce the associated hazards. In fact, physical wires cause the problem of signal point of failure. If the wire is damaged, all data transfer through it is paralyzed. In critical applications, this can have dangerous outcomes, especially in targeted attacks.

WSNs are mainly designed for surveillance purposes. They can be deployed in many fields such as military, automotive, agriculture, medicine...[Li et al., 2011]. Recently, industry has given WSN applications of monitoring a great deal of attention. Nowadays, they use sensor networks to monitor their machinery for maintenance scheduling. The sensors deployed to survey the system/component will provide data to assess the health, diagnose the system, and estimate the RUL. Yet, if this data is inaccurate, the prediction based on it will not be relevant. The dependability requirements (discussed in Section 2.5.4) need to be considered before the network starts running. Thereby, they can provide accurate data for RUL prediction and maintenance scheduling. Despite the existence of many dependability solutions in WSNs, these solutions are not always applicable. As sensors have restricted computational capabilities, solutions are often application-oriented.

As illustrated in Figure 3.16, and before starting the predictions, a WSN dependability study needs to be taken into consideration. As good predictions rely on real data, it is obvious that the first step is ensuring a reliable source of information. Once the provided

1. http ://www.beanair.com/success-stories/sahara-2-project

information are complete and correct, we will only need a robust health assessment model for good quality predictions.

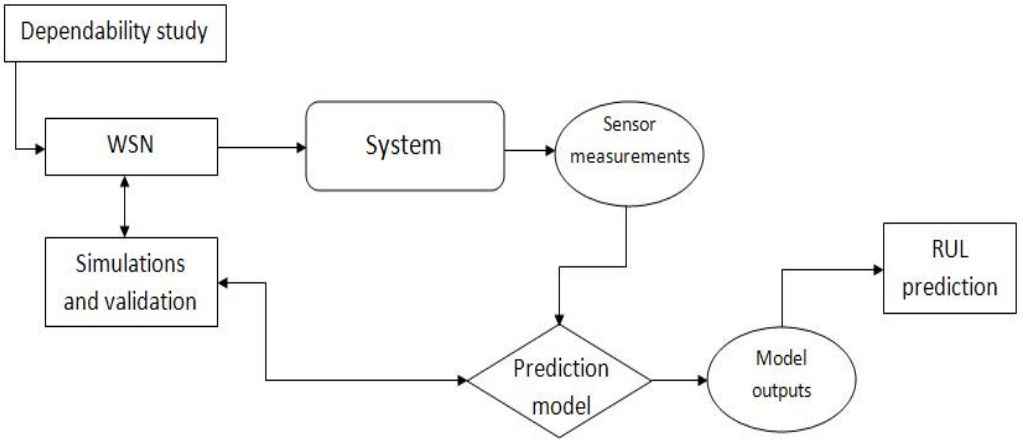


FIGURE 3.16 – General flowchart of CBM implementing a WSN.

WSNs are deployed for the data acquisition purposes. This acquired data will later be used in the health assessment process. Therefore, its quality has a clear impact on the model's outputs. For this reason, we focus our attention on improving the network reliability. In this thesis, we study three different network reliability attributes, which are enumerated hereafter.

1. **Network topology :** studying the network topology consists first in identifying the number of sensors to be used in the monitoring activity. This number depends on the number of targeted surveillance spots, the radio range of the sensors, and the wanted coverage rate. This study essentially aims at finding the optimal combination between the number of sensors and their location in a way that maximizes coverage range and minimizes energy consumption.

2. **Energy efficiency :** as sensor nodes are battery-powered devices, the critical aspect to face is how to reduce the energy consumption of the nodes, so that the network lifetime can be extended. In addition to network topology, the energy consumption can be optimized via the implemented routing protocol.

3. **Fault tolerance :** it is the property that enables a system to continue operating properly when a failure takes place in (or one or more faults within) some of its components. This essentially requires no single point of failure, a fault isolation mechanism, preventing the fault from propagation, and availability of the network.

The current state of the art in PHM is mostly concerned with improving the models results by focusing on the analysis step. Assuming the completeness and correctness of sensory data would only create a huge gap between the model building and model validating steps. This gap can be filled by ensuring the validity of the assumptions. This can be realized if we focus more on the observation phase of the CBM flowchart, as highlighted in Figure 3.17.

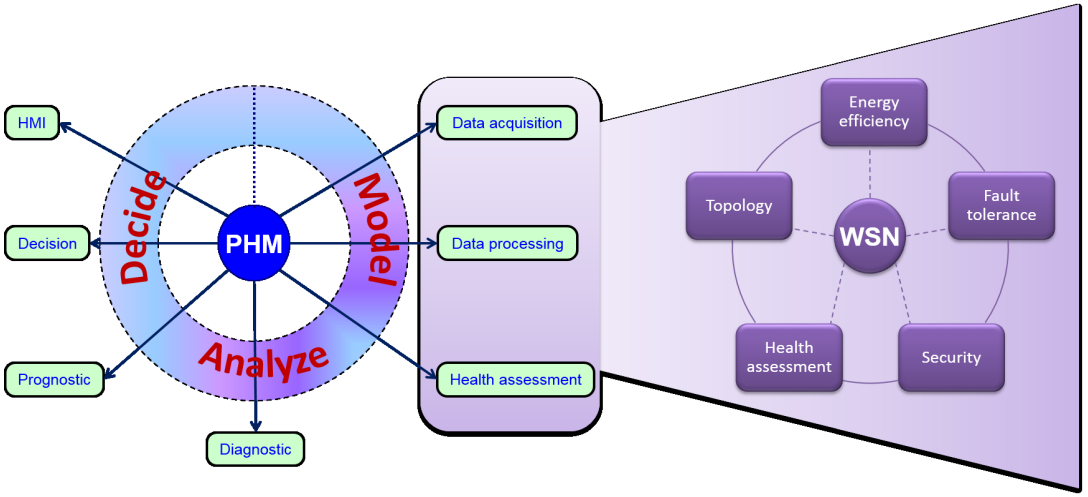


FIGURE 3.17 – CBM steps with WSN monitoring.

3.4/ CHALLENGES

Although many models have been developed in the PHM research area, there are many aspects that still need deep studying in order to provide more health assessment. How to use data fully ? How to consider operating conditions in health assessment and RUL prediction ? How to allow multiple interactions while building a model ? All these questions still need answers.

Data-driven models are designed to reduce model complexity and enhance real-time maintenance. For this reason, they only provide general predictions for a population of identical units ; this makes prediction process easier and faster.

In the literature of PHM, it is very common that the causes of a failure are limited to the values of monitored parameters. Other factors, although responsible of failures, seem to be neglected and overlooked. Although Condition Monitoring (CM) data reflect online monitoring, they do not replace reliability data. In fact, CM data provide measurements informing about a single component state at a specific moment. A failure does not only consider a single parameter (pressure, humidity...), it is a consequence of many factors (component age, different failing component...).

Reliability data, informing about all these factors, give a bigger picture of the failing process. We are not neglecting the importance of CM data. But, while CM data provide information for short-term prediction, reliability data are able to extend these predictions until next maintenance window. The complete neglection of operating conditions, operating age, and interactions between failures can only limit the application of developed models to real machines. Operating conditions are never the same, they change all the time. If the model is unable to consider these changes, then it is unable to produce reliable estimations. Furthermore, if we observe two similar components with different operating ages and operating under similar conditions, we will notice that they will not fail at the same time. Operating age definitely has an influence on time to failure. Even a failure can accelerate or provoke another one.

Another issue to face while performing health assessment, is censored data. Many plants do not allow their system to run to failure. Components are often replaced before they actually fail. As a result, the real time to failure is not kept record of. The performed preventive maintenance is mistaken for failure time, and RUL prediction is based upon

that time. The value of RUL is critical for maintenance scheduling. In other words, the less accurate is the prediction, the less reliable is the maintenance schedule.
Maintenance scheduling is the reason behind the entire PHM process. Yet, once accomplished, the maintenance actions are not considered in the model. And generally, the related component is considered "as good as new". It is very important to consider the effects of maintenance actions in the prediction model, at least to evaluate the model efficiency and study the new failure behavior after the maintenance being performed.
What also drew our attention are the assumptions upon which the models rely. To the best of our knowledge, none of the previous research work has questioned the availability, safety, and security of data. It is generally assumed that :

– Sensory data is available and there is no data loss.
– Sensor network is reliable.
– There is no fault in sensors.
– There is no constraint of energy consumption.

Unfortunately, the assumptions mentioned above in no way reflect a real life situation. The application of Wireless Sensor Networks (WSN) is very critical. First of all, the sensors' size is very small. So they have very small batteries with limited disposable energy. If the communication in the network does not consider this limitation, the sensors will quickly consume all the energy they have and be dropped. Thus, the information will no longer circulate in the network. Still, an energy efficient WSN will not stop some nodes from being dropped. This means that the network has to be fault tolerant in order to be able to pursue its functionalities in case of any sudden events (sensor loss, interferences...). Besides, like all wireless networks, WSN can be hacked. Competitors and hackers can steal information, change data, cause damage to the system... Data circulating in the network need to be secured against such attacks.
Many research works have been done in WSN reliability field. But every application has its own features, and generalized solutions do not always solve the problem.

So far, all research work is limited to the condition monitoring layer, the health assessment layer, and the prognostic layer of the Open System Architecture for Condition-Based Maintenance OSA-CBM [Thurston, 2001, Niu et al., 2010]. This architecture is illustrated in Figure 3.18.

In this thesis, we put the sensor module layer under the spotlight and study the data acquisition layer more closely. Doing so, we face the incompleteness of data packets and their impact on the PHM process. For this, we first study the reliability of the network to reduce the amount of packet loss, and then perform health assessment with incomplete information.

3.5/ CONCLUSION

Condition-based maintenance is an important tool for modern plants to optimize their maintenance schedule. An appropriate schedule is reflected by the economical benefits. In order to ensure that the developed degradation model guarantees the expected precision, it is important to focus on accuracy in the early steps pf the process, like health assessment. Building a degradation model depends on key issues, such as model complexity, model strengths, and the amount of available information. Data-driven models

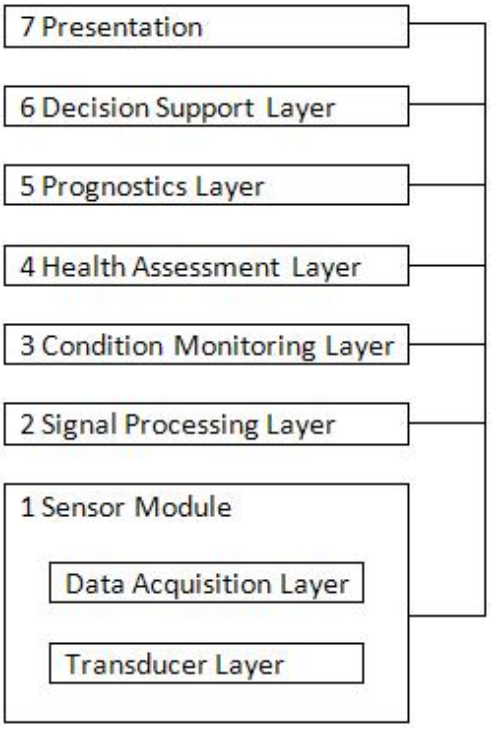


FIGURE 3.18 – Open System Architecture for CBM.

seem to be promising due to their advantages, such as the low complexity.
To the best of our knowledge, current research works utilize either independent sensors or sensors that are connected via physical wires for the data acquisition step. For some industrial systems, the use of wires gives rise to some complications. Wiring is sometimes too difficult that is almost impossible, which means, either we should adopt a wireless sensor network, or loose in terms of precision by placing the sensors a bit further from the target.

In the case where the industrial system is monitored by a wireless sensor network, data loss becomes highly probable, which has an important impact on the quality of predictions. In such a situation, the prognostic model is expected to maintain its robustness to the unpredictable lack of information. Considering the actual challenges of real-life applications, condition-based maintenance needs to be oriented in a way to meet the expectations of modern plants. As existent research works suppose that data is complete, we believe that previous solutions are unsuitable for wireless sensor networks monitoring. In the next chapters, we propose solutions to reduce the amount of data loss in the network, and to improve the quality of estimations when the monitoring data is incomplete.

II

CONTRIBUTION

4

Resiliency in Distributed Wireless Sensor Networks

In a monitoring activity, the sensor nodes are placed on/nearby the monitoring target to collect measurements of relevant parameters for health assessment (such as temperature). This helps assess the system's current state of health, diagnose the degree of its severity, and extrapolate the result in the future to estimate when the system is more likely to fail. The goal from this is to schedule maintenance activities in a way that avoids system failure and unnecessary shutdowns. Clearly, on-line measurements are crucial for an efficient Prognostics and Health Management (PHM). Consequently, the Wireless Sensor Network (WSN) used for the monitoring needs to be dependable, essentially by avoiding failures.

The network reliability can rely on retransmission and redundancy mechanisms. Since packet transfer consumes the highest amount of energy in the network, the retransmission based reliability does not respect the energy constraints of WSNs. When redundant paths are used, a packet is transmitted in multiple copies using different routes as a backup plan in case one of the routes fails [Al-Wakeel et al., 2007, Dijkstra, 1974, Mojoodi et al., 2011]. However, this solution results in unnecessary transmissions and therefore does not improve energy consumption in WSNs.

In the context of extending the network's lifetime, a possible solution is to maintain a minimum number of sensor nodes in an active mode [He et al., 2012b, He et al., 2012d, Kasbekar et al., 2011a]. Although this seems to solve the energy problem, other issues arise :

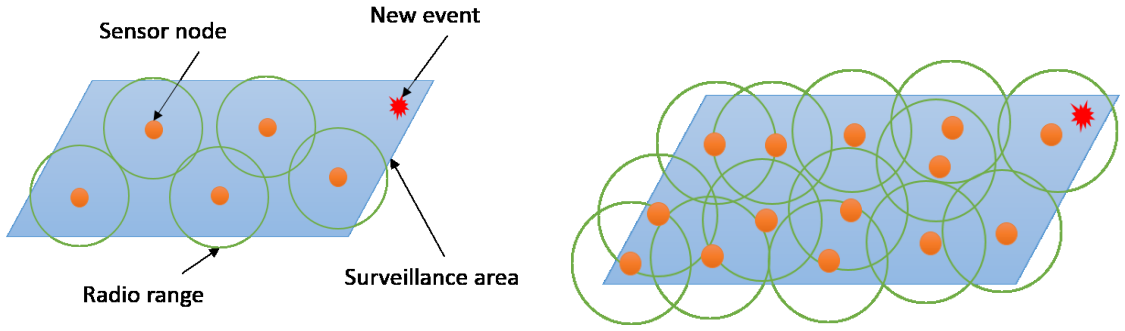
– How can we ensure a minimum coverage rate ?
– How can we reduce the loss of data ?
– How can we avoid unnecessary packet forwards ?

In this chapter,we present a distributed algorithm for resiliency in WSNs, which preserves the overall energy and takes into account the questions above mentioned.

4.1/ Coverage rate in wireless sensor networks

The coverage rate can be defined as the amount to which a region is covered, or in this case monitored. Deploying sensors with insufficient coverage can result in unreliable outputs. Therefore, the coverage rate is one of the most important factors in WSNs

[Kashi et al., 2012].



(a) Small coverage rate (b) High coverage rate

FIGURE 4.1 – An illustration of network coverage.

In figure 4.1, an example of the network coverage problem is illustrated. We can see in Figure 4.1(a) that the occurrence of the new event was not detected due to the low coverage rate, as opposed to Figure 4.1(b). With more events being undetected, the outputs of the network will not be reliable. In order to attain reliability in WSNs, sensing coverage and sensing level need to be considered. The sensing coverage refers to the integrated sensing area which is monitored by at least one sensor node. As for the sensing level, it refers to the number of sensor nodes being able to detect a new event when it takes place. [Choi et al., 2009] argue that existing node scheduling schemes focus on the minimum sensing level for the coverage problem and neglect the fault tolerance issues. In one hand, the minimum sensing level is an NP-complete problem. On the other hand, it cannot be preserved when nodes start to fail. Therefore, the authors propose the Fault-tolerant Adaptive Node Scheduling (FANS) algorithm, which efficiently handles the degradation of the sensing level. The algorithm designates a set of backup nodes for each active node. If the latter fails, the predesignated set of backup nodes activate themselves to replace it and to restore the lowered sensing level. FANS requires a small number of backup nodes and a small amount of control messages. [Chen et al., 2009] study fault tolerant out-of-band monitoring for WSNs. They aim at placing a minimum number of monitors in a sensor network in a way that all sensor nodes are monitored by k distinct monitors, and each monitor serves at most w sensor nodes. The authors first prove that this problem is NP-hard and then propose three algorithms providing near optimal solutions.

Battery level, broken links, and communication failures have an impact on the Quality of Service (QoS) of WSNs. This leads to consequences varying from disturbing the traffic in the network to completely interrupting it. [Geeta et al., 2013] propose an Active node-based Fault Tolerance using Battery power and Interference model (AFTBI) to identify the faulty nodes in WSNs. Fault tolerance against low battery power is assured through a hand-off mechanism where the faulty node selects the neighbor with highest battery level and transfers all the services towards it. To reduce interference signal, a dynamic power level mechanism is introduced, where the power of a node is adjusted automatically with regards to its current state (active or asleep). Simultaneous transmissions can be avoided if the nodes are only allowed to transmit data within a time slot. [Lee et al., 2008] tackle the same problem by identifying and isolating the faulty sensor nodes in the network. Sensed data is compared among neighbors to determine its accuracy. Once the predetermined fault threshold is reached, the node in question is isolated from the diagnosis

process ; a faulty node can be included in data transferring but not data sensing. Transient faults in communication and sensor reading are tolerated by using the time redundancy mechanism. The drawback of this solution is that faults are assumed to be only related to the sensing activity, excluding other sources of failure.

Energy in WSNs can also be preserved through lifetime optimization. [Kasbekar et al., 2011b] leverage prediction to prolong the network lifetime, by exploiting temporal-spatial correlations among the data sensed by different sensor nodes. Based on Gaussian Process, the authors formulate the issue as a minimum weight submodular set cover problem and propose a centralized and a distributed truncated greedy algorithms (TGA and DTGA). They prove that these algorithms obtain the same set cover. Lifetime optimization using knowledge about the dynamics of stochastic events has been studied by [He et al., 2012c]. The authors presented the interactions between periodic scheduling and coordinated sleep for both synchronous and asynchronous dense static sensor network. They show that the event dynamics can be exploited for significant energy savings, by putting the sensors on a periodic on/off schedule. [He et al., 2012d] design a polynomial-time distributed algorithm for maximizing the lifetime of the network. They proved that the lifetime attained by their algorithm approximates the maximum possible lifetime within a logarithmic approximation factor. [Zhang et al., 2014] presented a stochastic sensing algorithm to reduce energy consumption through node scheduling. They used data correlation between nodes to reduce error rate by adjusting duty cycle of faulty sensors. Their algorithm conserves 60 % of energy as compared to other solutions, while confining sensing error within specified error tolerance. [He et al., 2012a] use actors to allocate spare sensors to sensor-deficient regions or to relocate sensors from sensor-abundant regions to sensor-deficient regions. They introduce a baseline centralized greedy algorithm for sensor allocation, where global sensor information is communicated to obtain the optimal solution. The works cited here focus on a periodic schedule for turning the sensors on and off.

Data collection delay and reliability need to be considered in scheduling algorithms for WSNs. [Zhang et al., 2012] claim that existing algorithms have not solved these two problems effectively. The authors propose the Fault-Tolerant Scheduling (FTS) algorithm, where each sensor node detects the environment and generates some sensing data at regular intervals. The algorithm helps surviving network malfunction by switching the parent of a sensor node to its backup parent. The simulation results show that FTS has a short data collection time and high fault tolerance. [Feng et al., 2011] considered the problem of efficient data aggregation in WSNs by putting in place amendment strategies in case of failures. Their solution needs local information to repair the aggregation tree and automatically reschedules nodes for interference free aggregation after the amendment. [Cheng et al., 2013] present STDG, an efficient data gathering scheme based on matrix completion. STCDG takes advantage of the low-rank feature instead of sparsity, thereby avoiding the problem of having to be customized for specific sensor networks. They exploit the presence of the short-term stability feature in sensor data, which further narrows down the set of feasible readings and reduces the recovery errors significantly. Furthermore, STCDG avoids the optimization problem involving empty columns by first removing the empty columns and only recovering the non-empty columns, then filling the empty columns using an optimization technique based on temporal stability.

To preserve the overall energy in the network, sensor nodes are on a periodic schedule where they are switched on only when the sensing level is decreased. An optimal schedule needs to take nodes failure rate and the elapsed run-time into consideration. When

the failure rate is small, wakening the nodes too often would only waste energy. As we go further in time, nodes start to exhaust their energy supply and this is when they start to fail. A combination a node failure rate and elapsed time would give us a better indication of the optimal nodes wakening schedule.
Maintaining the sensing level considerably reduces the amount of packet loss, yet it does not completely prevent its occurrence. A sudden node failure will result in the permanent loss of the held data packet, unless a redundancy mechanism is put in place. In the context of reducing energy consumption, the redundancy solution should be avoided and replaced by other solutions which do not include unnecessary packet transmission.
In this thesis, we propose a fully distributed algorithm for resiliency in wireless sensor networks where the number of awake nodes is kept to a minimum that ensures the network coverage. Each sensor copies its data on its neighbor, and these copies are only retrieved in case of a sensor failure. This algorithm is detailed in the following.

4.2/ THE PROPOSED ALGORITHM

To cope with fault tolerance and data survivability, a fully distributed algorithm is presented and theoretically analyzed. Our algorithm seeks to cover data loss by maintaining a necessary set of working nodes and recovering failed ones when needed. We suppose that we are in the case of high density networks, and not all nodes participate in the network's service. Some nodes are in an idle state because their targets are actually covered by working sensors. We consider that these idle sensors wake up periodically to check for eventual node failures and therefore ensure their targets' coverage. In case of failures, they decide to switch to active mode and therefore initiate the recovery process to retrieve the data of the failed nodes. However, during the network's service, how can we handle the case where two (or more) sleeping nodes would realize at the same time that the working neighbor is down ?

Indeed, two neighboring sensor nodes may be elected at the same time, and the recovery process of two neighboring nodes may be the same. This algorithm aims at filling this gap by proposing an efficient node failure recovery scheme in order to allow sensor networks to gracefully degrade in performance instead of failing unpredictably.

In the following, we first focus on the legitimate state formulation and next, we present the algorithm which consists in only three rules and we give the correctness proofs.

4.2.1/ PROBLEM FORMULATION

Let $G = (V, E)$ be the graph modeling the sensor network, with $|V| = n$ and $|E| = m$. We assume that sensor node identifiers are unique. We recall that a sensor node identifier is unique if and only if $i.Id \neq j.Id$ holds for each $i, j \in V (i \neq j)$. A sensor node can be in one of these three states : *failed, working,* or *probing*. Every node i in the network has to maintain the following data structure :

- D_i : the sensed data by node i. Each time a node updates D_i, it sends/replicates the newly sensed data to/on its neighbors.
- P_i : the parity information on node i. It is the result of the combination of the replicated information of its neighbors.

We considered two different scenarios for the parity information. In the first scenario, there are no memory constraints. Each new data is saved on a different memory register, and we used the SUM function for data collection. In the contrary, all information must be saved on the same memory register when it comes to the second scenario. So, we used the XOR function to preserve memory space.

Let $T = t_1, t_2, ..., t_k$ be the set of monitoring targets to be covered and $S = 1, 2, ..., n$ the set of sensor nodes. Each target in T has to be covered by at least one sensor node in S. We call Γ_u the set of neighbor- sensors of target $t_u, 1 \leq u \leq k$. Each neighbor-sensor $j \in \Gamma_u$ is capable of monitoring the target t_u, formally :

$$\forall j \in \Gamma_u : ds(t_u, j) \leq R_s, \Gamma_u \subseteq S, t_u \in T,$$

where $ds(t_u, j)$ denotes the distance between t_u and sensor j.

Let N_i be the initial set of neighbors of node i and $d_i = |N_i \setminus \Gamma_u(i) \setminus i|$, the number of its working neighbors. As the number of failures goes up with time, we let d_i^* be the dynamic number of alive neighbor nodes. We denote by D^k the set of $d_i + 1$ replicas of data D_i. Also, we denote by $s\ (D^k)$ the sensor node to which data-replica D^k is assigned, for $1 \leq k \leq d_i + 1$ and by $\hat{s}\ (D^k)$ the elected sensor node who recovers D_i if node i fails. The data are replicated on different nodes (space exclusion, see Lemma 1) since the goal is to achieve data survivability even if some node failures occur in the network.

We say that a sensor node i is independent if

$$i.state = working \wedge (\forall j \in \Gamma_u(i))(j.state = sleeping \vee probing \vee failed)$$

and that i is dominated if

$$(i.state = sleeping \vee probing) \wedge (\exists j \in \Gamma_u(i))(j.state = working)$$

The legitimate state (let us denote it Σ) of the network is then expressed as follows :

$\forall i \in V : i.state = failed$
$\Rightarrow ((\exists \hat{s}, \hat{s}' \in \Gamma_u(i))(\hat{s}.state = \hat{s}'.state = working) \Rightarrow (\hat{s}(D_i^k) = \hat{s}'(D_i^k)))$
In other words, each data loss is recovered by at most one working sensor node.

4.2.2/ THE ALGORITHM

When a sleeping node wakes up, it sends a *probe-request* message to check if there exist working nodes in its vicinity. If no working nodes, it recovers the lost data of the failed node and starts to operate in the active mode ; otherwise, it sleeps again. Nodes are initially in the sleeping mode. Each node sleeps for an exponentially distributed time generated according to a probability density function (PDF) $f(t) = \lambda e^{-\lambda t}$, where λ is the probing rate of the sensor node and t denotes its sleeping time duration.

Upon detecting an eventual failure, a probing node i updates its actual probing rate λ_i by taking into account the dynamic number of alive neighbors $d_i^* : \lambda_i^{new} \leftarrow \lambda_i \,.\, \frac{d_i}{d_i^*}$. Then, a new sleeping period is generated by using the new computed parameter λ_i^{new} according to the PDF function : $f(t) = \lambda^{new} e^{-\lambda^{new} t}$. The following notations are also given for the predicates of node i

- $W(i)$: working neighbor : $\exists j \in \Gamma_u(i), i.state = working$
- $W^*(i)$: working neighbor with lower Id : $\exists j \in \Gamma_u(i), j.state = probing \wedge i.Id > j.Id$

- $F(i)$: failed neighbor : $\exists j \in \Gamma_u(i), j.state = failed$
- $P^*(i)$: probing neighbor with lower Id : $\exists j \in \Gamma_u(i), j.state = probing \wedge i.Id > j.Id$

The proposed algorithm uses the following three rules :

$rule1$:

if $(i.state = probing \wedge (P^*(i) \vee W(i)))$ **then**

  if $P^*(i)$ **then**

    $\lambda_i^{new} \leftarrow \lambda_i . \frac{d_i}{d_i^*}$

  end if

  $i.state \leftarrow sleeping$

end if

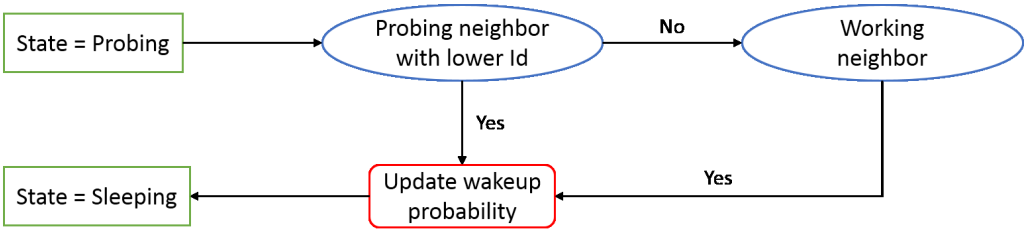


FIGURE 4.2 – Algorithm rule 1.

$rule2$:

if $i.state = probing \wedge (\neg W(i) \wedge \neg P^*(i) \vee F(i))$ **then**

  if $F(i)$ **then**

    if memory constraint **then**

      $D_i \leftarrow P_z \underset{k \in N_z \backslash \Gamma_u(i), k \neq j}{\oplus} D_k$ (*$F(i) = F(z) = j$*)

    else

      $D_i \leftarrow D_k, k \in N_z \backslash \Gamma_u(i), k \neq j$ (* k is chosen randomly*)

    end if

  end if

  $i.state \leftarrow working$

end if

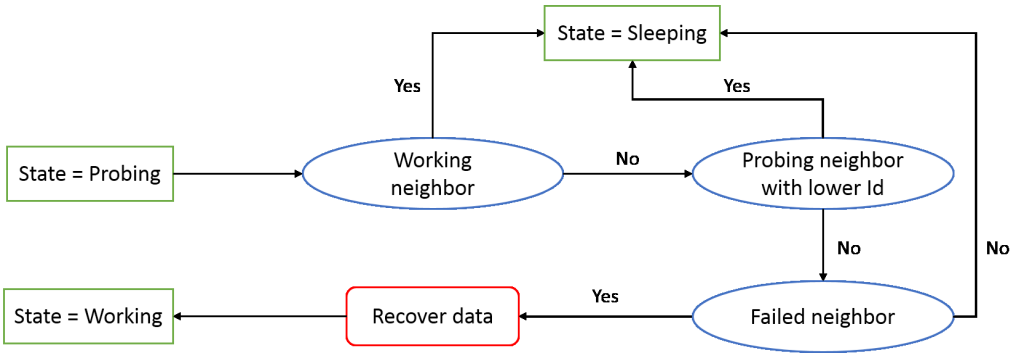


FIGURE 4.3 – Algorithm rule 2.

$rule3$:
if $(i.state = working \wedge W^*(i))$ **then**
$i.state \leftarrow sleeping$
end if

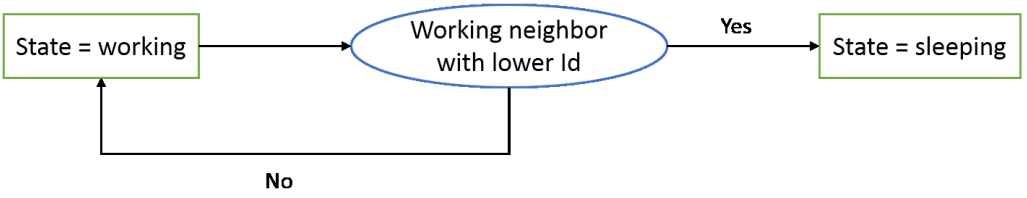


FIGURE 4.4 – Algorithm rule 3.

4.2.3/ CORRECTNESS PROOFS

In this section, we detail properties of our fault tolerant algorithm, and express its validity/convergence. We assume that links are trustworthy/flawless and lossless.

Lemma 1. *A sensed data D_i is guaranteed to survive in the presence of d_i permanent faults if and only if $s(D_i^k) \neq s(D_i^{k'})$, for $1 \leq k, k' \leq d_i + 1$.*

Démonstration. If d_i nodes fail, then there is $s(D_i^u), 1 \leq u \leq d_i + 1$ which did not fail, and therefore D_i^u will be recovered successfully from $s(D_i^u)$ since there are $d_i + 1$ copies of D_i assigned to $d_i + 1$ different nodes. However, if there is a sensor node $s(D_i^k), 1 \leq k \leq d_i + 1$, such that $s(D_i^k) = s(t_i^u) = s^*$ and s^* fails, then neither D_i^k nor D_i^u can be recovered successfully. □

Lemma 2. *If at most d_i neighbors crash down for any sensor node $i \in V$ in the network, then the algorithm is valid and resists to eventual node failures.*

Démonstration. The proposed algorithm is based on replication scheme with space exclusion. Thus, according to Lemma 1, each data is replicated d_i+1 times onto d_i+1 distinct sensor nodes. We have at most d_i node failures at the same time. So at least one copy of each data is recovered from a fault free node. □

Lemma 3. *If a node changes to the working state by $r2$, then it remains in its state and will never execute a rule again until an eventual failure takes place.*

Démonstration. Let i be a sensor node that executes $r2$. According to the preconditions of all rules, node i can execute only rule $r3$ in the next round. However, in order to do so, one of its neighbors would have to switch to $working$ state following $r2$. This is impossible as long as node i is in the $working$ state. Thus, node i will never execute a rule again. If node i is down, it remains in its state (fail-stop failure). □

Lemma 4. *If a sensor node is enabled by rule r2, then each one of its neighbors will execute at most one more rule until their next wake-up/probing, and this rule will be r1.*

Démonstration. Let i be a node that executes $r2$. When node i changes to $working$ state, all its neighbors are either in $sleeping$, $probing$, or $failed$ state. So, we have three possible scenarios : i) neighbors in $sleeping$ state : there is no conflict in this case. ii) neighbors with $probing$ state : these neighbors have a higher Id than i. iii) failed neighbors will remain in their state until their recovery. □

Lemma 5. *Every sensor node is either independent, dominant, or failed.*

Démonstration. From the point of view of node i, we have three scenarios :

- if node i is in the $working$ state and is not $independent$, then i may execute rule $r3$.
- if node i is in the $sleeping \vee probing$ state and is not $dominated$, then node i may execute rule $r2$.
- if node i is in the $failed$ state, then node i will remain in its state until its recovery.

□

Lemma 6. *When a node is not failed $\vee$ sleeping, it can make at most two moves.*

Démonstration. By Lemma 3 and Lemma 4, each rule can be executed at most once by a node. Hence, the only case a node makes two moves is when it executes $r3$ then $r2$ with a $working$ state. □

Theorem 1. *With respect to the legitimate state Σ of the network, the proposed algorithm converges within 2n moves.*

Démonstration. This follows from Lemma 1 to Lemma 6. □

4.2.4/ MESSAGE COMPLEXITY ANALYSIS

In the following, we give an Upper-Bound of the actual number of probe/reply messages exchange during the network's lifetime task.

Theorem 2. *The number of probe/reply messages involved by the algorithm is at most :*

$$O\left(n\ m \times \max_i \frac{t_i^{R_i}}{\Delta_i}\right), 1 \leq i \leq n$$

where, n is the number of nodes, m is the number of virtual communication links, $t_i^{R_i}$ is the reliable lifetime of node i and Δ_i is the smallest sleeping period time of node i.
This bound is attainable.

Démonstration. **1.** The reliable lifetime $t_i^{R_i}$, of the node $i, 1 \leq i \leq n$ for a specified reliability R_i, starting the mission at age 0, is computed as follows :
$R_i = 1 - F(t_i^{R_i}) = e^{-\lambda t_i^{R_i}} \Rightarrow ln\ R_i = -\lambda t_i^{R_i} \Rightarrow t_i^{R_i} = -\frac{1}{\lambda}\ ln\ R_i$
This is the lifetime during which the sensor node i will be functioning successfully with a reliability of R_i.
According to node's sleeping periods subdivisions of the time, we have :
$0 = t_o < t_1 < t_2 < ... < t_k = t$. Let $\Delta_p = [t_{p-1}, t_p[, 1 \leq k$ denote the p^{th} sleeping period time. Since the number of failures goes up, the sleeping time period decreases with

time. This implies that the probing process of node i costs at most $O\left(\frac{t_i^{R_i}}{\Delta_i}\right), 1 \leq i \leq n, \Delta_i = min\Delta_p, 1 \leq p \leq k$. In addition, for each probing message issued from node i, we may have the corresponding reply messages from its working neighbors. This cost is at most $O(|N_i|)$. Therefore, from the point of view of node i, the number of probe/reply messages is at most $O\left(|N_i| \times \frac{t_i^{R_i}}{\Delta_i}\right)$.

Finally, summing up for the whole n sensor nodes, the algorithm's message cost is at most

$$O\left(\Sigma_{i=1}^{n}|N_i| \times \frac{t_i^{R_i}}{\Delta_i}\right) \leq O\left(n\ m \times max_i \frac{t_i^{R_i}}{\Delta_i}\right), 1 \leq i \leq n$$

2. To see that this bound is really attainable, consider a linear chain graph of only two sensor nodes s_1 and $s_2(n = 2)$. We need to orchestrate the involved communications between these nodes in time. Assume that s_1 is working and s_2 is the passive state. If $t_1^{R_1} = t_2^{R_2}$ (s_1 and s_2 start functioning and fail at the same time), then the whole number of probe-message issued from s_2 is $\frac{t_2^{R_2}}{\Delta_2}$, where Δ_2 is the constant sleeping time period of s_2. Since both s_1 and s_2 have the same life for which nodes will be functioning successfully, node s_1 will reply for each probing message issued from s_2. As a result, the whole number of involved probe-request/reply message before the failure of s_1 and s_2 is $n\ m \times \frac{max_i t_i^R}{min_i min_j \Delta_{ji}} = 2 \times \frac{t_2^{R_2}}{\Delta_2}$

□

4.3/ SIMULATION ENVIRONMENT

The study of WSNs' performances is complex. Although the theoretical study gives an idea of the expected performance, it is not sufficient to predict the behavior of a network. So we need simulation, using an affirmed simulator, adopted by the scientific community. Given its popularity and efficiency, we choose NS2 for the study of the performance.

NS2 [1] is an open-source event-driven simulator designed specifically for research in computer communication networks. It provides a graphical environment to better observe what happens during the simulations. Its NAM (Network Animator) tool visualizes the topology, packets routing between nodes, etc.. This simulator can implement a comprehensive architecture with several nodes. It also implements several mobility models to model mobile entities. In addition, a large number of simulation models and communication protocols have been implemented as well as several models of energy consumption. AS results, NS2 provides a trace file that records all data and statistics throughout the simulation, a file containing the graphic animation of the network that displays the progress of the simulation, the moving nodes, the routing of data packets and even signaling. Figure 4.5 shows a diagram of simulation steps with NS2.
NS2 consists of OTcl and C++. The C++ objects are mapped to OTcl handles using TclCl. To run a simulation, a user needs to define a network scenario in a Tcl Simulation script, and feeds this script as an input to an executable file ns. During the simulation, the packet flow information can be collected through text-based tracing or NAM tracing. After the simulation, an AWK program or a perl program can be used to analyze a text-based trace

1. http ://www.isi.edu/nsnam/ns/ (last consulted on 09/16/2015)

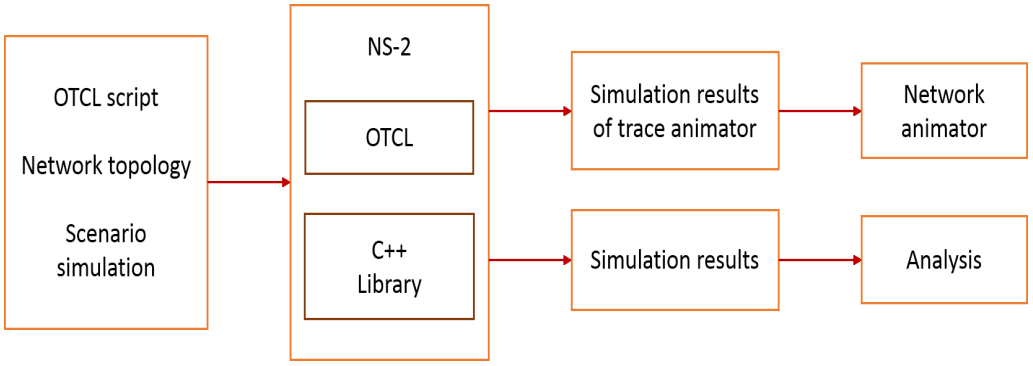


FIGURE 4.5 – Simulation steps using the NS-2 simulator.

file. The NAM program, on the other hand, utilizes a NAM trace file to replay the network simulation using animation.

Simulation using NS2 consists of three main steps. First, the simulation design is probably the most important step. Here, we need to clearly specify the objectives and assumptions of the simulation. Secondly, configuring and running simulation implements the concept designed in the first step. This step also includes configuring the simulation scenario and running simulation. The final step in a simulation is to collect the simulation result and trace the simulation if necessary.

Two of the most important aspects in a network simulation are debugging and compilation of simulation results. Debugging refers to a procession of removing compilation and run-time errors in both C++ and OTcl domains. This chapter provides guidelines and necessary commands for debugging. Although originally designed to facilitate the understanding of network dynamics, NS2 tracing could also be useful in the debugging process. NS2 supports two types of tracing. Variable tracing records the changes in value of a variable (in most cases in a file), while packet tracing stores the details of packets passing through network checkpoints (again in most cases in a file) [Issariyakul et al., 2008].

4.4/ SIMULATION PARAMETERS

In this section, we give a brief description of the tested metrics to evaluate the algorithm.

4.4.1/ NETWORK'S LIFETIME

This metric, widely used in the WSN literature, is computed to reflect the time span from the network's initial deployment to the end of activity manifested by the sensor nodes running out of energy to send data packets. It is the amount of time that a Wireless Sensor Network would be fully operative. Some definitions link the network lifetime to the first loss of coverage. As such, network lifetime can alternatively be defined as the time until the first node dies. However, losing a node could mean that the network could lose some functionalities, and not all of them. It is also possible to use a different definition, in which when some nodes die or run out of battery power, the network does not necessarily reach the end of life. Whenever other network nodes could be used to capture desired information or to route information messages to their destination, the purpose of the network is

maintained. The easiest to capture indicator of this metric is the maximum per-node load, where a node's load corresponds to the number of packets sent from or routed through the given node. Clearly, the network setup that minimizes the maximum node load is the one that will ensure the maximum network lifetime.

4.4.2/ RECOVERY FAILURE

Generally speaking, data loss is a crucial metric to evaluate a routing protocol. It is useful to know whether a protocol is able to minimize packet loss, in order to decide if it is performant.
One of the goals of the proposed algorithm, is salvaging the lost data packets. When a node fails (from energy exhaustion for example), its data is salvaged from a neighboring nodes (where a copy is stored). This metric will help us test how well the algorithm is able to recover from losing data packets. For each data recovery attempt, we will record how many times recovery has not been properly performed. An attempt coincides with a failing node, and a new node carrying the data transferring activity by recovering the lost packet (due to failure). Recovery failure reflects the percentage of data salvaging failure in proportion to the total number of attempts. The lower this percentage is, the better performance we have.

4.4.3/ COVERAGE RATE

The coverage rate is an important metric to evaluate the performance of a WSN. It reflects the amount to which a region is covered, or monitored in this case. Deploying sensors with insufficient coverage can result in unreliable outputs. It also measures the network's ability to meet its monitoring obligations. More explicitly, the higher the coverage ratio, the better the ability of the network to fulfill its surveillance task.
This is metric is of high importance to us cause it ensures the continuity of the CBM process. In other words, if a zone is not covered, no data can be provided for health assessment. The coverage rate has a percentage value that reflects the fraction of the network that was monitored during the simulation.

4.4.4/ NUMBER OF MESSAGES

The number of messages circulating in the network can give a clearer view of the extra cost induced by the algorithm.
A sparse network would result in a small number of exchanged messages. Increasing the density of the network (the number of nodes) will give us a clear view of the algorithm's complexity. More specifically, if the number of messages increases exponentially when the number of nodes grows, this would mean that the algorithm has a high complexity.

4.4.5/ NUMBER OF WAKE-UPS

Wake-ups are designed for on-demand communication scheme. A node remains in the inactive state unless it is solicited.

In the proposed algorithm, the nodes wake up following a probability function. This function takes into account the time that has elapsed and the number of failures of nodes in the network. The goal of the function is to find a reasonable relation between the number of times an inactive node wakes up and the number of nodes failure in the network.
Evaluating this number will give us a comprehensive view of the relationship between the density of the network and the probability of wake-ups in one hand, and between the failure rate and the number of wake-ups on the other.

4.5/ SIMULATION RESULTS

In this section, we discuss some results through simulations. We consider a flat grid topology of 10 by 10 *i.e.* 100 monitoring zones. We vary the number of sensors between 200 and 1600 nodes. Since sensors are uniformly distributed in the monitoring area, the density of sensors at each zone varies between 2 and 16.

The performance evaluation considers five aspects : (i) Network lifetime evolution ; (ii) Failure rate : that is the ratio of information recovery attempts that did not succeed ; (iii) Effective monitoring time : this measure is related to the time between the death of the active node in a monitoring zone and its replacement ; it is expressed in (%) ; (iv) Total number of messages ; and (v) Number of awakenings per inactive sensors.

Our simulations are performed in two different settings : the first setting sets a low wake-up rate but enough to keep the monitoring time ratio higher than 50 % in almost all configurations ; while the second setting considers a wake-up rate four times higher than the rate in the previous setting. This allowed us to reach monitoring time ratios up to 90 %.

These two settings were put in place in order to test and compare two different solutions for data collection. In the first scenario, we assume that there is no constraint related to memory capacity of sensor nodes. Therefore, each sensor is able to save data received from all nodes in its neighborhood on a different memory register (this method is called SUM). As for the second scenario, we aim for preserving the memory space. Thus, we suppose that each node has one memory register that is available to save all information received from all its neighboring nodes ; every new data packet is added in the register using the function XOR. We can notice that the second method for data saving (XOR) is highly sensitive to any neighboring node failure. In fact, the failure of a neighbor induces a corruption of the calculation of the data needed for the recovery process. For this reason, the coverage rate needs to be high enough for this solution to work. Consequently, the XOR method is only implemented in the 4x version of our simulation settings.

4.5.1/ WAKE-UP RATE = 1X

In this section, only the SUM method is implemented. The nodes in the network are designed to fail randomly. This failure rate varies from 0 % to 8 % by a pitch of 2 %.

In Figure 4.6, we can observe that the network's lifetime increases in an almost linear manner. For a small network, the number of times a sensor node receives a wake-up message is higher comparing to the same number when the network is larger. Therefore, the more nodes are participating in the coverage process, the less energy is consumed per node and the overall energy in the network is preserved.

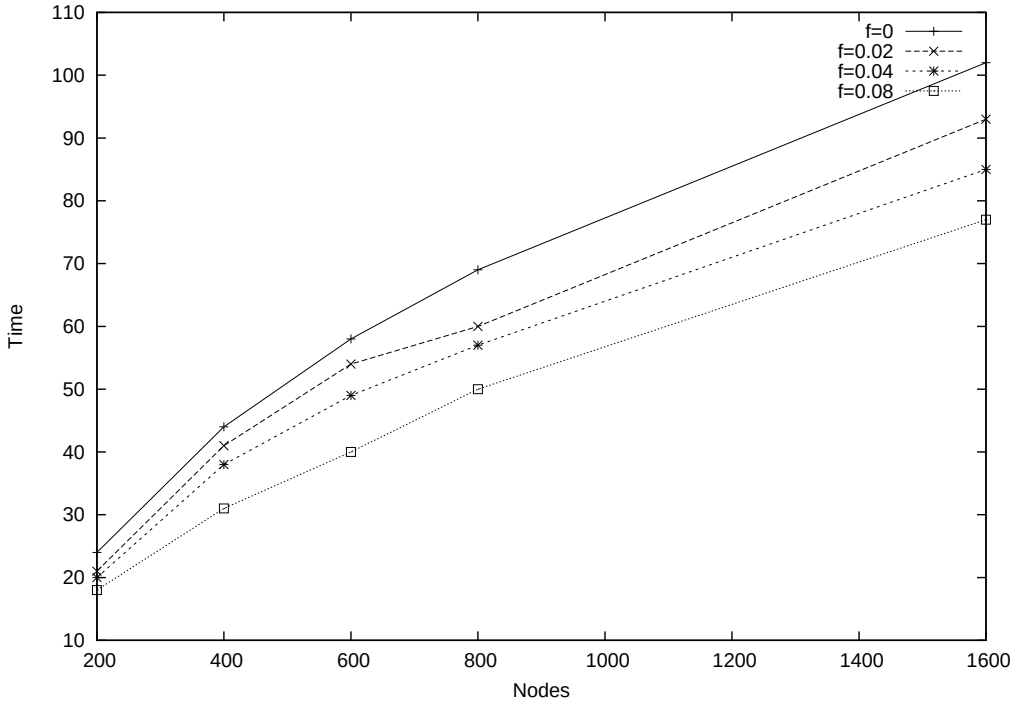


FIGURE 4.6 – Network's lifetime.

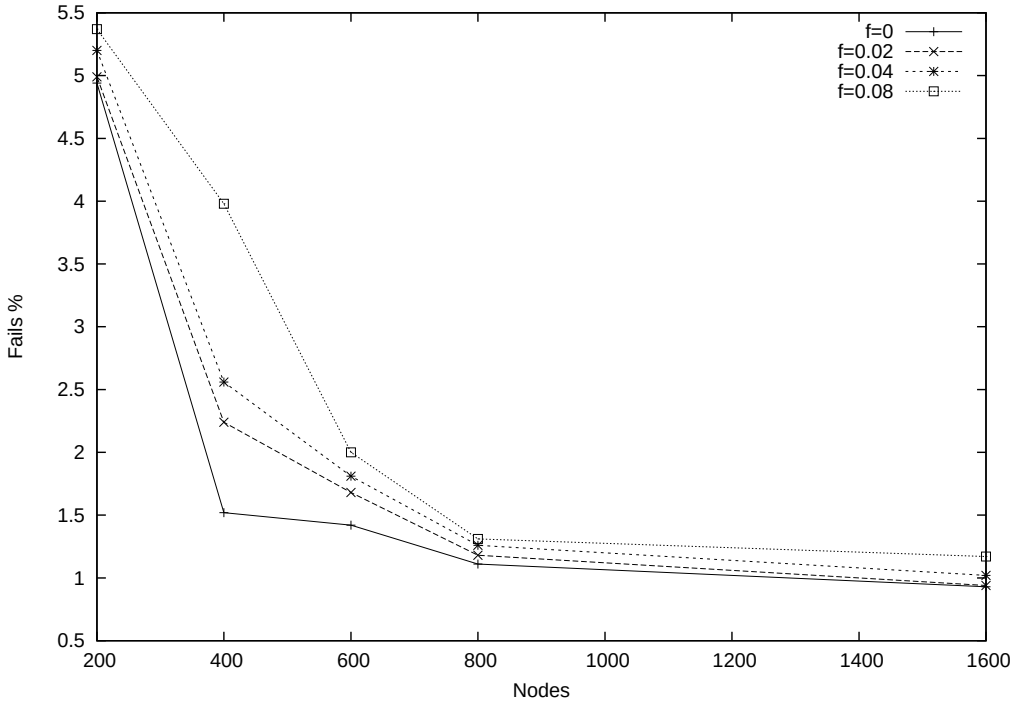


FIGURE 4.7 – Failure of the recovery process.

The overall energy level has an impact on the recovery process. In fact, when the energy level starts to go down, the number of nodes able to cover a given area is reduced. The remaining nodes will receive an increasing number of wake-up messages and when they

fail, the number of replacements is continuously decreased. Eventually, some zones will no longer be covered. Thus, data recovery rate increases when the network is larger. Even though failures are less impacting for a dense network, the failure rate is low even for a small network (see Figure 4.7).

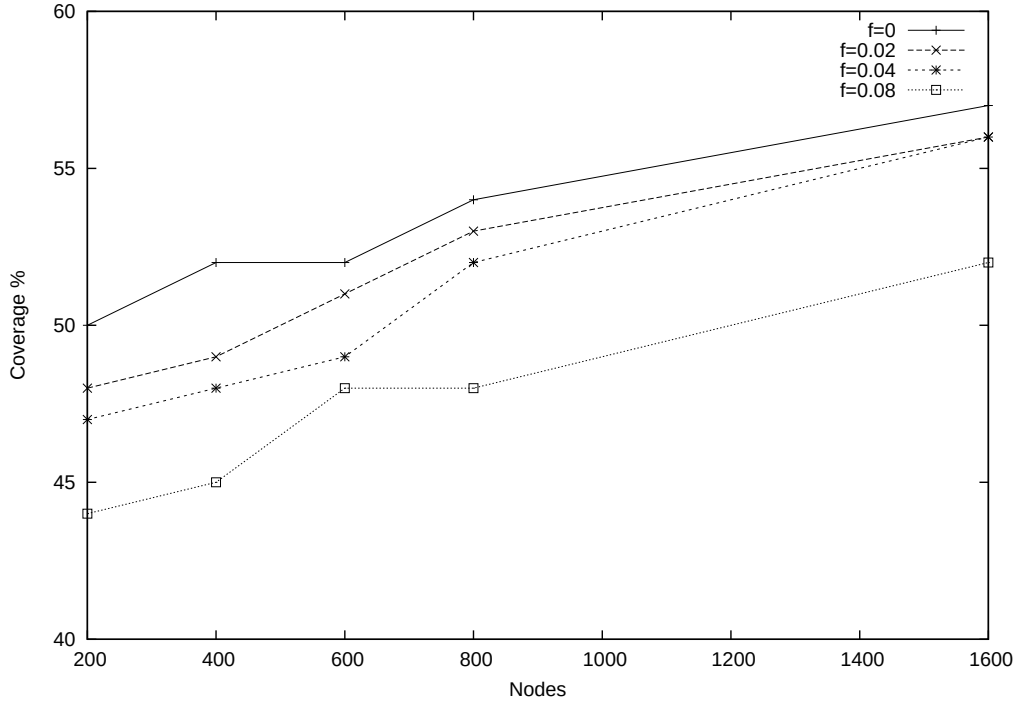


FIGURE 4.8 – Coverage rate.

The coverage rate (illustrated in Figure 4.8) is more successful when the density of the network grows. Nevertheless, it remains between the values of 45 % and 55 % depending on the settings. Indeed, as discussed above, the number of wake-up messages in the network is higher with the growth of nodes number and therefore more energy is dissipated. On the other hand, when the number of nodes is small, even though the number of wake-up messages is reduced, nodes fail faster as the number of node replacements is small. Consequently, there is no huge difference in the coverage rate. Still, a dense network guarantees a better coverage rate.

The total number of exchanged messages in the network will only grow with the increased number of nodes in the network as shown in Figure 4.9. In fact, each node will copy its sensed data onto each one of its neighbors. So when the number of nodes increases, the number of messages increases accordingly. The growth of messages number is linear rather than exponentially. This reflects the low complexity of the algorithm.

The number of wake-ups per node illustrated in Figure 4.10 follows a logarithmic form. Sensor nodes periodically wake up to verify if there zone is being covered by an active node. The wake-up rate follows a probability function that is updated considering node failure. So, the number of these messages highly depends on the number of nodes failure. When the probability of node failure increases, the number of nodes in the network is decreased and thus the total number of wake-ups.

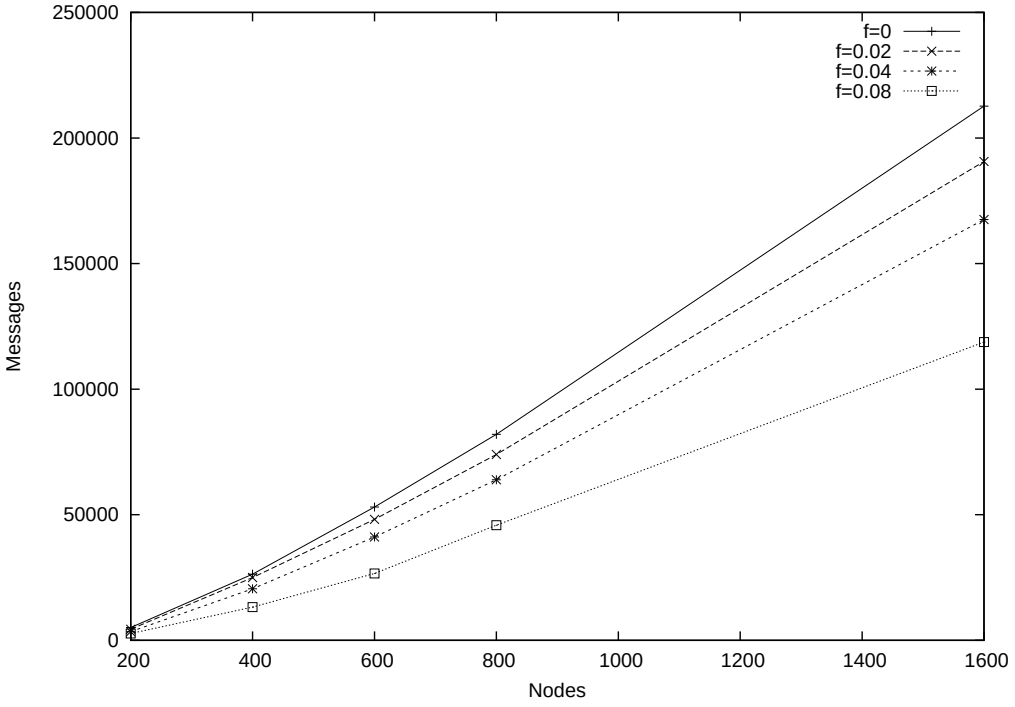


FIGURE 4.9 – Number of total messages in the network.

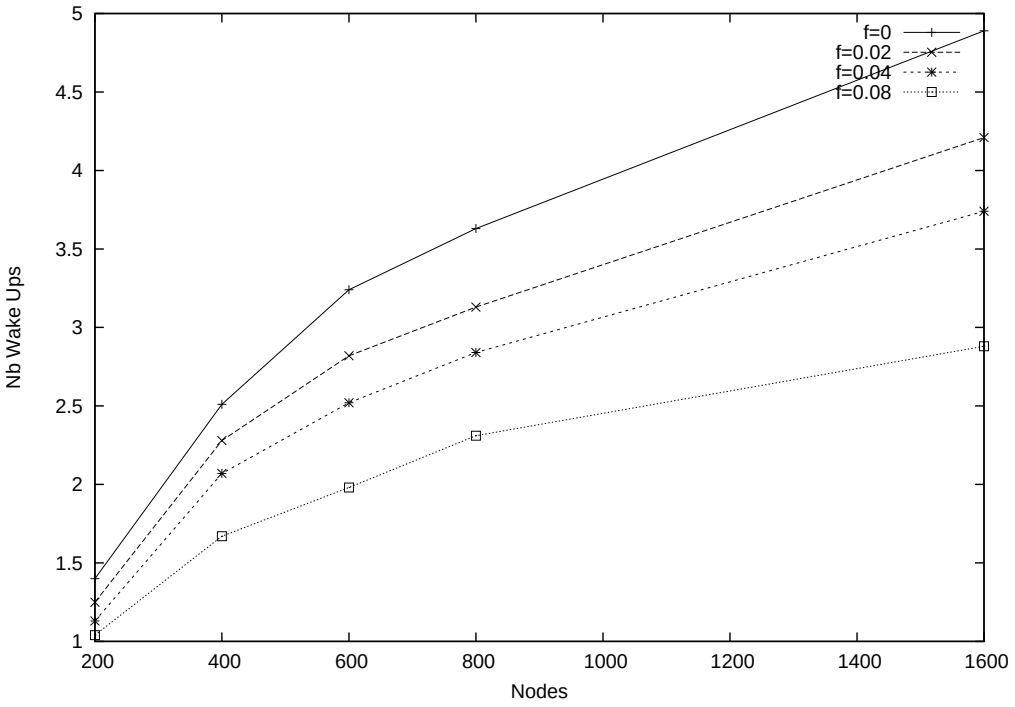


FIGURE 4.10 – Number of total wake-ups in the network.

4.5.2/ WAKE-UP RATE = 4X

In this section, both of SUM and XOR methods are tested. Since all the curves are similar, except for failure of the recovery process, only the figure corresponding to the latter illustrates the comparison between both methods. Nodes failure rate are fixed to 0 % and 8 % only (the two extreme cases from the previous configuration).

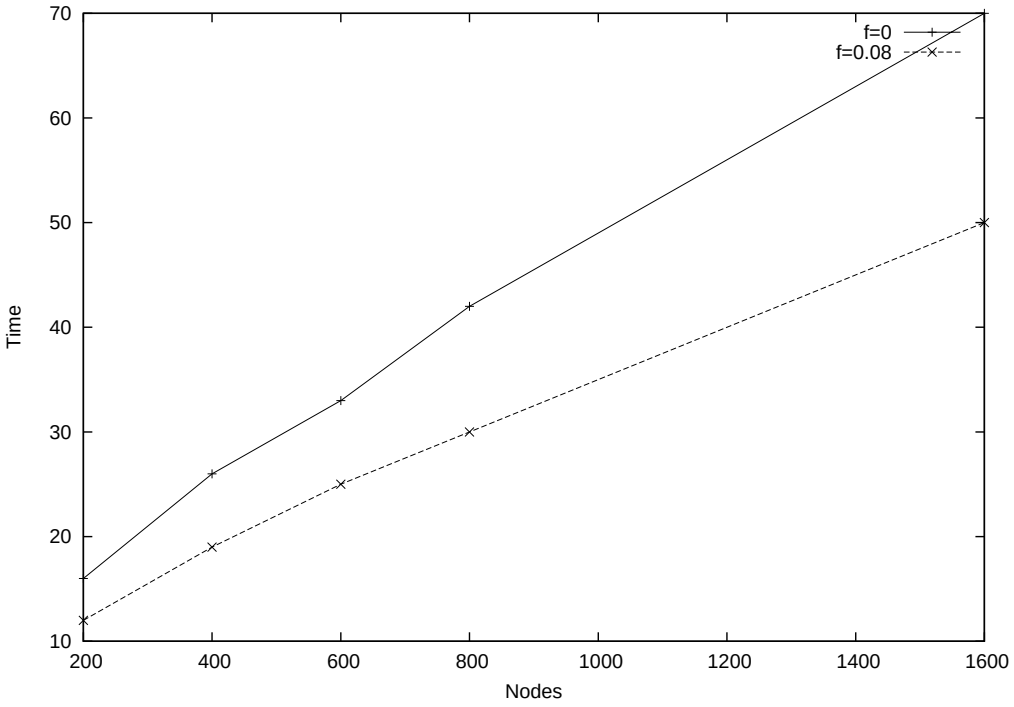


FIGURE 4.11 – The network's lifetime.

Comparing to the previous configuration, the network's overall lifetime has decreased. Considering that the wake-up rate here is 4 times more frequent, it is normal that network consumes more energy in this setting (see Figure 4.11).

Nevertheless, the different zones coverage rate illustrated in Figure 4.13 was considerably and understandably improved. The failure of the recovery process in Figure 4.12 remains very low with the absence of memory constraints, and even lower comparing to the previous configuration. In the contrary, the XOR function appears to be highly sensitive to node failure. When the failure rate reaches 8 %, the recovery failure jumps by 30 % for a small network and 15 % when the network is dense.

The total number of exchanged messages is considerably higher than the number in the previous configuration, and this is due to the increased number of wake-up messages. The algorithm also improves the overall energy consumption by only maintaining a necessary set of nodes in the active mode. The rest of the node wake up randomly to check their area and ensure that coverage is performed. This random function is optimized by updating it accordingly to the nodes failure rate.

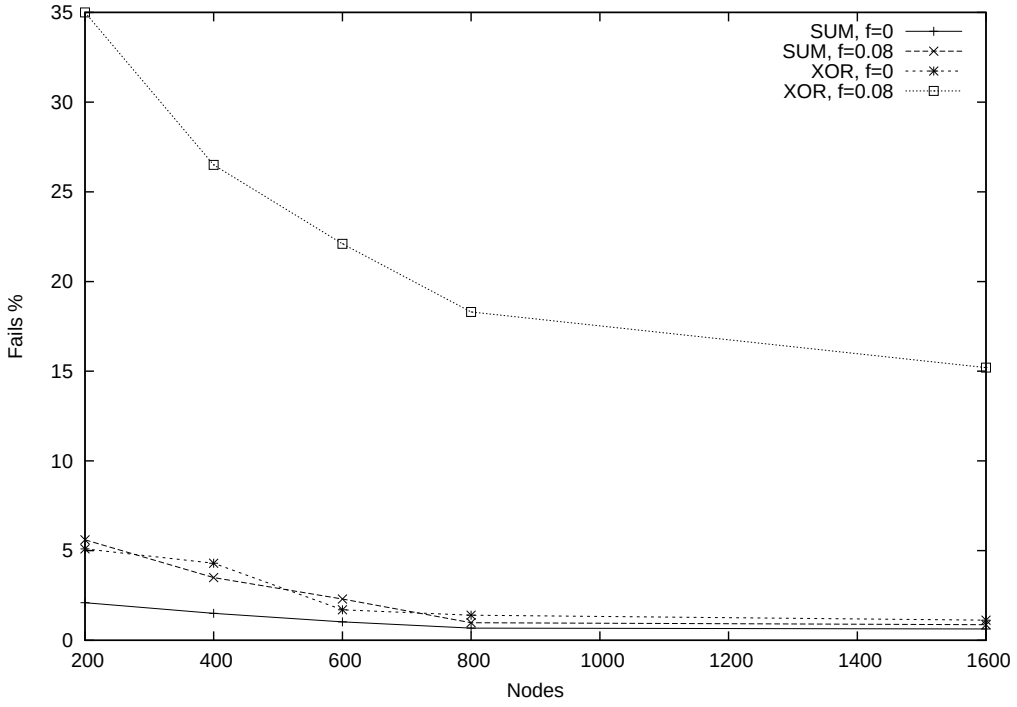


FIGURE 4.12 – The failure of the recovery process.

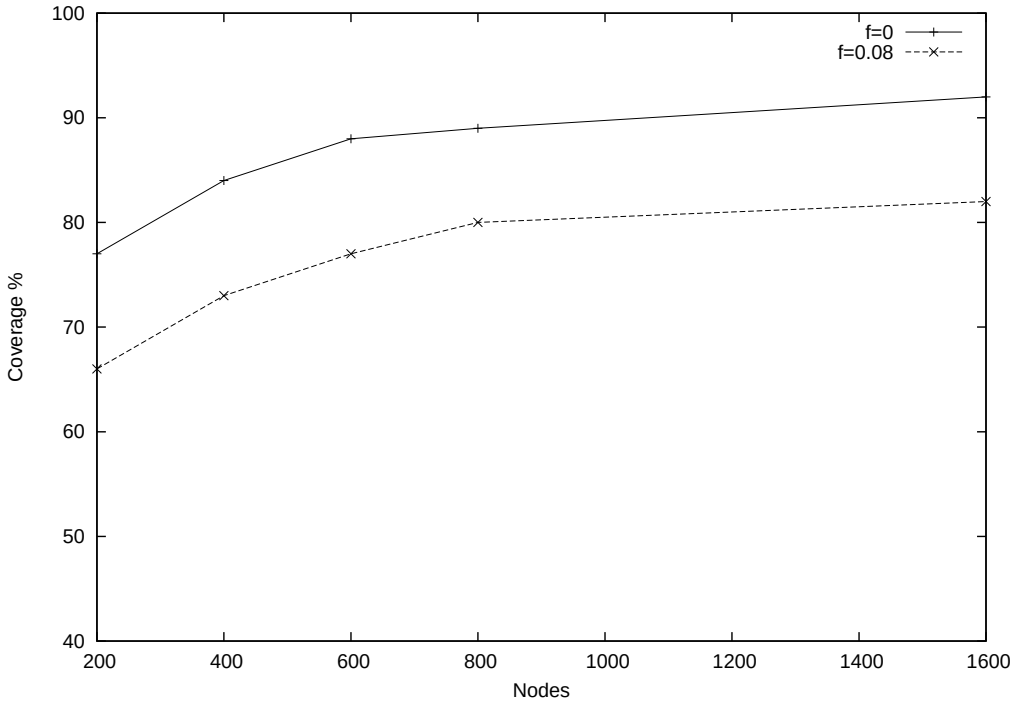


FIGURE 4.13 – The coverage rate.

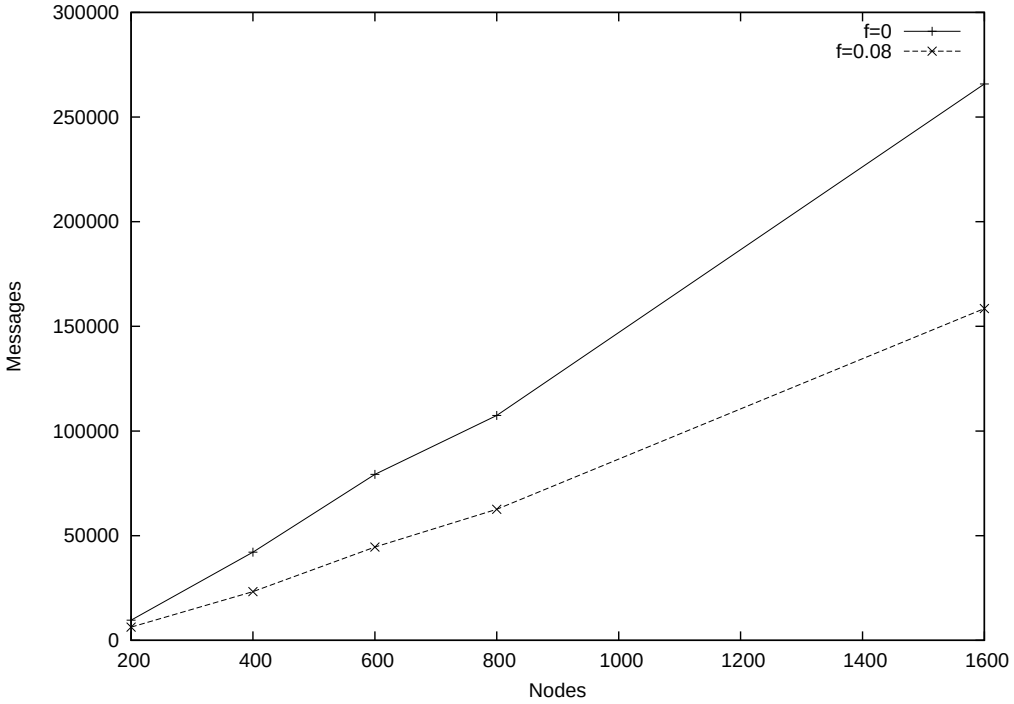


FIGURE 4.14 – The number of total messages in the network.

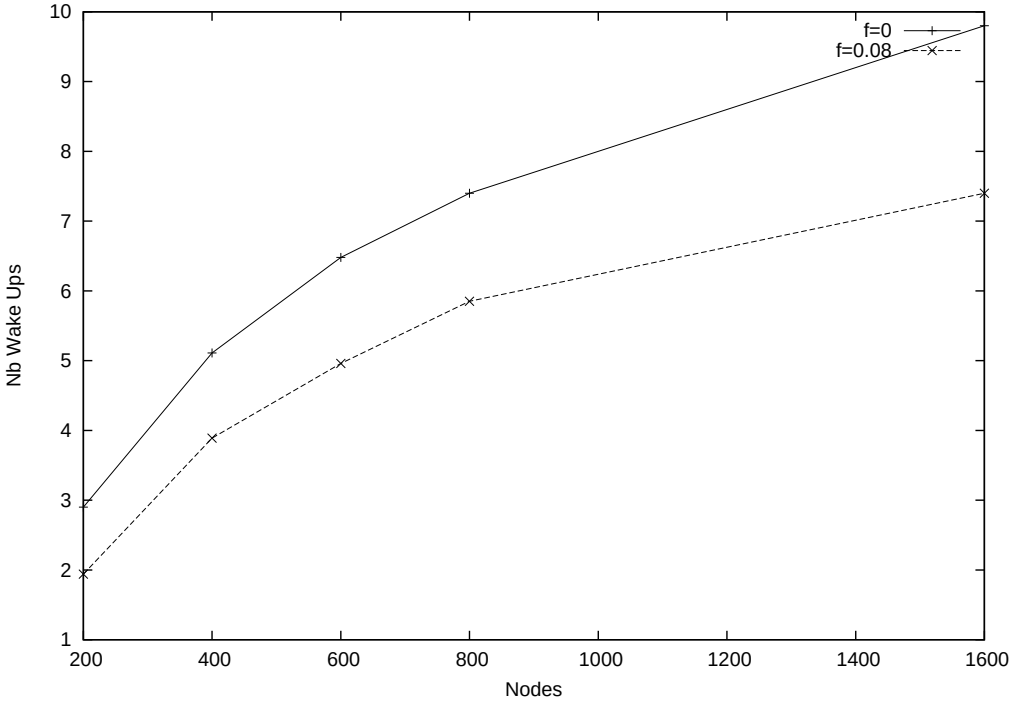


FIGURE 4.15 – The number of total wake-ups in the network.

4.6/ CONCLUSION

In this chapter, we proposed a fully distributed algorithm that seeks to reduce both energy consumption and data loss rate in the network. The algorithm only maintains a minimum

necessary sensors at the working mode. The working nodes are able to ensure coverage while eliminating interference. This means that for a surveillance distance, which is equal to the sensor's radio range, one and only one sensor is ensuring the monitoring activity.

The algorithm also recovers failed sensor nodes when needed. At the start of their activity, each sensor node copies its data on its neighbors, using two different assumptions : (i) in the first one, we suggest that there is no memory constraint and each new information is copied on a different register, and (ii) in the second one we put in place a memory constraint and use the XOR function to add a new data to the common memory register for all data. We also tested two different configurations, where the wake-up rate is 4 times more frequent from one configuration to the other. The performed simulations showed that a more frequent wake-up rate helps improve the quality of the recovery process. Even though the absence of memory constraints facilitates the recovery process, this rate was maintained below 35% for a small network and around 15% for a dense one even in the presence of memory constraints. This algorithm also helps preserve the energy in the network by only maintaining a necessary set of sensor nodes in the active mode. The rest of the nodes wake up randomly to ensure that their area is covered by a sensor node. This random function is optimized by updating it according to the nodes failure rate.

We managed to exploit the benefits of the redundancy and retransmission mechanisms, all while eliminating the constraints related to the energy consumption in the network. With the proposed algorithm, the network's lifetime is more important, and the amount of lost data is reduced, yet not eliminated. In the next chapter, we propose a solution for performing health assessment with incomplete measurement at the processing unit.

5

HEALTH ASSESSMENT VIA THE RANDOM FOREST ALGORITHM

Health assessment is a cornerstone in the prognostics and health management process. Predicting the remaining useful life begins with identifying the system's current state of health. Based on the sensor measurements, the values of the monitoring parameters will be used in the process of categorization and differentiation between the different states the system can be in. This task is called classification.

In many areas of information science, finding predictive relationships from data is a very important task. Initial discovery of relationships is usually done with a training set while a test set and validation set are used for evaluating whether the discovered relationships hold. More formally, a training set discovers potential predictive relationships and a test set assesses the strength and utility of a predictive relationship.

This entire process takes success from the availability and correctness of the monitoring data. Nevertheless, in the context of wireless sensor network monitoring, the probability of losing data packets is very high. In Chapter 4, we presented an algorithm that minimizes data loss. Considering that data is never complete, we continue working on improving the quality of predictions with missing information. In this chapter, we focus on the adaptability of the health assessment algorithm to the changes in the online observations. For this, we investigate the performance of different network topologies regarding the completeness of data. We also perfor health assessment using the random forest algorithm and study the accuracy of the results when the on-line measurements are incomplete.

5.1/ MACHINE LEARNING

In machine learning, classification refers to identifying the class to which a new observation belongs, on the basis of a training set and quantifiable observations, known as properties. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Such algorithms can operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions. Different groups of machine learning algorithms are shown in Figure 5.1.

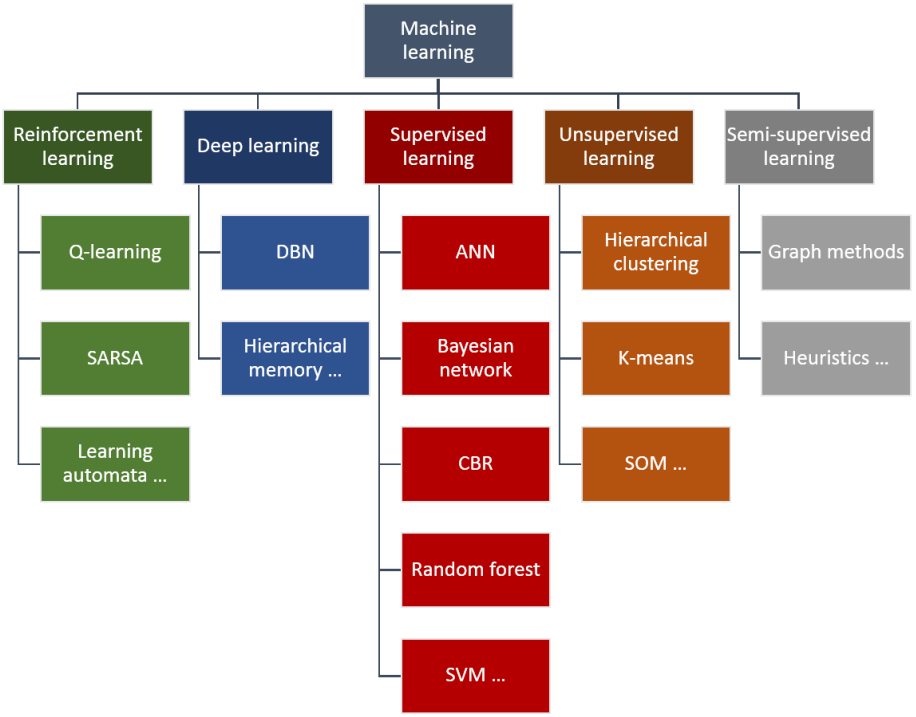


FIGURE 5.1 – Machine learning algorithms.

5.1.1/ REINFORCEMENT LEARNING

Reinforcement learning is a type of machine learning, and thereby also a branch of artificial intelligence. It is inspired by behaviorist psychology and concerned with how software agents ought to take actions in an environment to maximize the cumulative reward. Simple reward feedback is required for the agent to learn its behavior ; this is known as the reinforcement signal. It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance.

The environment is typically formulated as a Markov decision process (MDP). In the problem, an agent is supposed decide the best action to select based on his current state. When this step is repeated, the problem is known as a Markov Decision Process. The main difference between the classical techniques and reinforcement learning algorithms is that the latter do not need knowledge about the MDP and they target large MDPs where exact methods become infeasible.

Reinforcement learning differs from standard supervised learning in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected. Further, there is a focus on on-line performance, which involves finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).

There are many challenges in current Reinforcement Learning research. Firstly, it is often too memory expensive to store values of each state, since the problems can be very complex. Solving this involves looking into value approximation techniques, such as Decision Trees or Neural Networks. There are many consequence of introducing these imperfect value estimations, and research tries to minimize their impact on the quality of the solution. Moreover, problems are also generally very modular ; similar behaviors reappear often, and modularity can be introduced to avoid learning everything all over again. Hie-

rarchical approaches are common-place for this, but doing this automatically is proving a challenge. Finally, due to limited perception, it is often impossible to fully determine the current state. This also affects the performance of the algorithm, and much work has been done to compensate this Perceptual Aliasing.

5.1.2/ DEEP LEARNING

Deep machine learning is based on a set of algorithms that attempt to model high-level abstractions in data. It consists in learning representations of data, where an observation can be represented in many ways. Research in this area attempts to make better representations and create models to learn these representations from large-scale unlabeled data. Some of the representations are inspired by advances in neuroscience and are loosely based on interpretation of information processing and communication patterns in a nervous system, such as neural coding which attempts to define a relationship between the stimulus and the neuronal responses and the relationship among the electrical activity of the neurons in the brain.

A main criticism of deep learning concerns the lack of theory surrounding many of the methods. In fact, deep learning methods are often looked at as a black box, where most confirmations are done empirically, rather than theoretically.

5.1.3/ SUPERVISED LEARNING

Supervised learning aims at training a model, on the basis of a known set of data (inputs) and their known corresponding responses (outputs), that generates predictions to new observations. For the training set, it is essential that each input has a know corresponding output. The data set is therefore called labeled. A supervised learning algorithm analyzes the training set and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations. The main steps for a supervised learning are : (1) Preparing data, (2) choosing an algorithm, (3) fitting a model, (4) choosing a validation method, (5) Examining fit and updating until satisfactory results, and (6) using fitted models for predictions.

Usually, these algorithms are fast and accurate. They are able to generalize by giving a correct result when new data is given in input without knowing *a priori* the target. However, over-fitting is a common problem. Over-fitting happens when the algorithm performs well on the training set and poorly with new observations. This means that the algorithm learned the data and not the underlying function. An other drawback of supervised learning is the computational complexity when large data sets are used for the training.

5.1.4/ UNSUPERVISED LEARNING

The problem of unsupervised learning is the problem of finding a hidden structure in unlabeled data. All the observations are assumed to be caused by latent variables, that is, the observations are assumed to be at the end of the causal chain. In practice, models for supervised learning often leave the probability for inputs undefined. This model is not

needed as long as the inputs are available, but if some of the input values are missing, it is not possible to infer anything about the outputs. If the inputs are also modeled, then missing inputs cause no problem since they can be considered latent variables as in unsupervised learning.

Unsupervised learning has advantages such as the model can be not provided with the correct results during the training. It can be used to cluster the input data in classes on the basis of their statistical properties only, and the labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

On the down side, since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.

Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to pre-process data.

5.1.5/ SEMI-SUPERVISED LEARNING

Semi-supervised learning falls between supervised learning (with completely labeled training data) and unsupervised learning (without any labeled training data) tasks and techniques and makes use of unlabeled data for training, typically a small amount of labeled data with a large amount of unlabeled data. Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human agent or a physical experiment. The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Semi-supervised learning is also of theoretical interest in machine learning and as a model for human learning.

5.2/ ENSEMBLE METHODS

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance that could be obtained from any of the constituent learning algorithms. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble refers only to a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

Ensemble methods could be either homogeneous or heterogeneous. In homogeneous methods, the same algorithm is used for the classification problem. It is the data set that is varied, as indicated in Figure 5.2. The algorithm is applied to each of the subsets and the final classifier is the combination of all the hypotheses.

In an heterogeneous method, different algorithms are applied to the same training set, as illustrated in Figure 5.3.

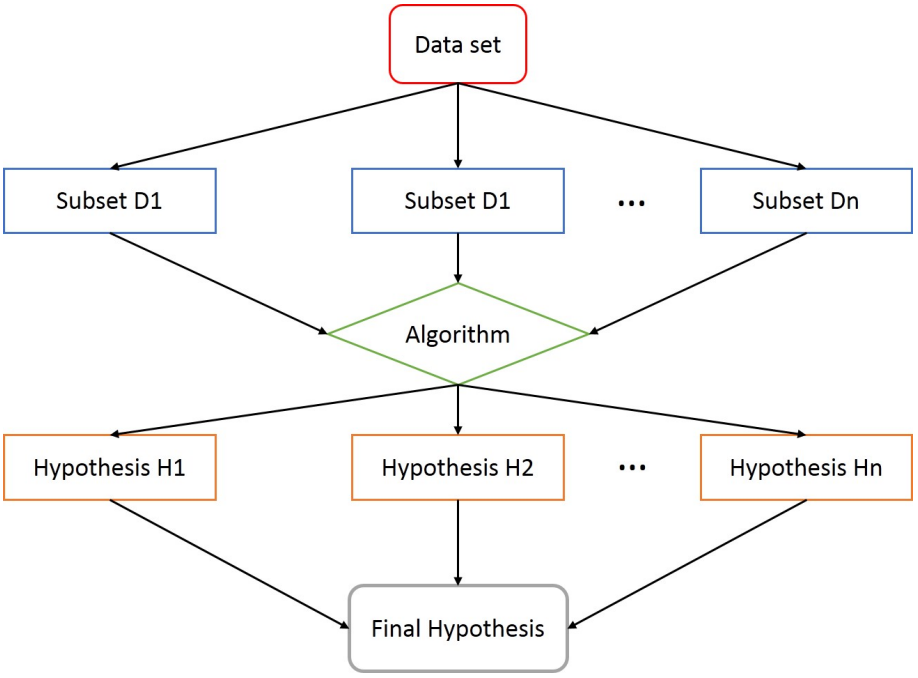


FIGURE 5.2 – Data set diversification in homogenous ensemble methods.

If the training space is large in proportion to the number of examples, several hypotheses with the same performance may be induced. The algorithm is then forced to choose one, that is probably not the best. An ensemble method solves this problem by averaging a number of hypotheses, thus reducing the overall error.

Nevertheless, ensemble methods have some constraints. While an homogeneous method would need a reasonable size of data for the training, an heterogeneous one might induce some conflicts of fusion of results, due to different degrees of accuracy of the different algorithms.

5.3/ THE RANDOM FOREST ALGORITHM

In ensemble learning, the results provided by different classifiers are combined to solve a particular computational intelligence problem. Many research works encourage adapting this solution to improve the performance of a model, or reduce the likelihood of selecting a weak classifier. For instance, [Dietterich, 2000] argued that averaging different classifiers' outputs guarantees a better performance than the weakest classifier among them. This claim was theoretically proven correct by [Fumera et al., 2005]. The fusion of multiple classifiers can even improve the performance of the best individual classifier, if particular hypotheses are taken into consideration [Tumer et al., 1996].

Two of the early examples of ensemble classifiers are Boosting and Bagging. In Boosting algorithm [Schapire, 1999], the distribution of the training set changes adaptively based on the errors generated by the previous classifiers. At each step, a higher degree of importance is accorded to the misclassified instances. At the end of the training, a weight is accorded to each classifier, regarding its individual performance and indicating its importance in the voting process. As for Bagging [Breiman, 1996], the distribution of the

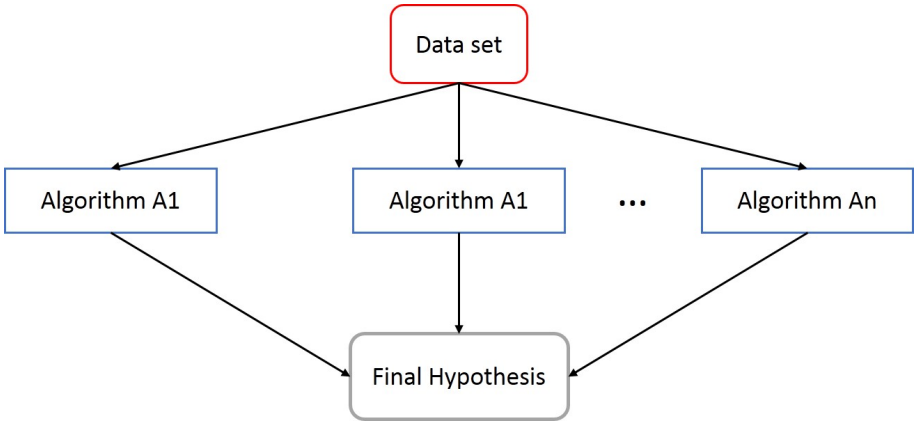


FIGURE 5.3 – Algorithm diversification in heterogeneous ensemble methods.

training set changes stochastically and equal votes are accorded to the classifiers. For both classifiers, the error rate decreases when the size of the committee increases.

In a comparison made by [Tsymbal et al., 2000] between the two previous algorithms, it is shown that Bagging is more consistent but unable to take into account the heterogeneity of the instance space. In the highlight of this conclusion, the authors emphasize the importance of classifiers' integration.
Combining various techniques can provide accurate results, as different classifiers will not behave in the same manner faced to some particularities in the training set. Nevertheless, if the classifiers give different results, a confusion may be induced [Kanemoto et al., 2013]. It is not easy to ensure reasonable results while combining the classifiers. In this context, the use of random methods could be beneficial. Instead of combining different classifiers, a random method uses the same classifier over different distributions of the training set. A majority vote is then employed to identify the class.

In this thesis, the use of Random Forest (RF) is proposed for industrial Health Assessment (HA), particularly in the context of devices being monitored using a Wireless Sensor Network (WSN). An example of a RF is a given in Figure 5.4.

The choice of RF was motivated by a number of factors. First of all, the injected randomnss in the algorithm gives the trees different starting points. In fact, when sensor nodes start to fail, certaint measurements will not arrive at the base station for processing. When the corresponding monitored parameter is needed at the root of the decision tree, the health assessment can not be performed (due to lack of information). the injected randomness will give the process a continuity using other trees in the forest (trees that need another parameter). Second of all, the feature selection step is included in the algorithm. In other algorithms used for health assessment (the PHM process in general), the feature selection step needs to be done aside. The RF algorithm includes this step in the training phase. This will be further detailed in Section **??**.

Health assessment is a key step for Remaining Useful Life (RUL) estimation. Based on the analysis and the predefined thresholds, the machine/component's health state is identified. Sensory data is reported periodically to monitor critical components. This data corresponds to measurements of monitoring parameters and is useful to assess the machine/component's condition. Each monitoring parameter has a threshold, once reached, the system is considered to be in the corresponding state.

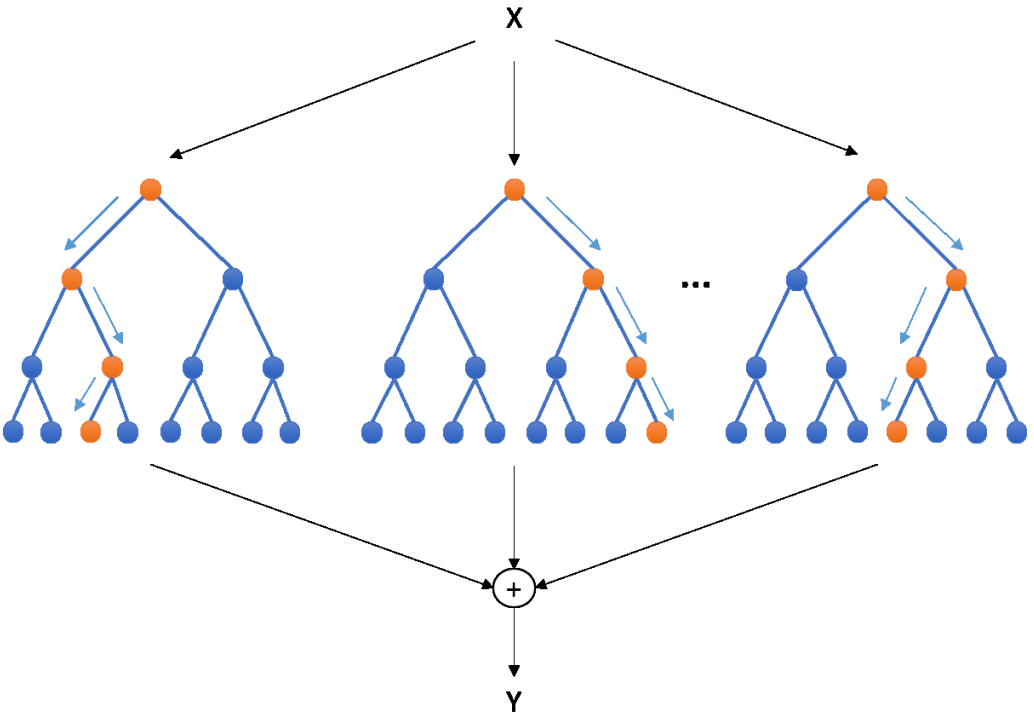


FIGURE 5.4 – An illustration of the random forest.

The RF algorithm is mainly the combination of Bagging [Breiman, 1996] and random subspace [Ho, 1998] algorithms, and was defined by Leo Breiman as "a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest" [Breiman, 2001]. This method resulted from a number of improvements in tree classifiers' accuracy.

This classifier maximizes the variance by injecting randomness in variable selection, and minimizes the bias by growing the tree to a maximum depth (no pruning). The steps of constructing the forest are detailed in Algorithm 1 and illustrated in Figure 5.5.

In a RF, the root of a tree i contains the instances from the training subset S'_i, sorted by their corresponding classes. A node is terminal if it contains instances of one single class, or if the number of instances representing each class is equal. In the alternative case, it needs to be further developed (no pruning). For this purpose, at each node, the feature that guarantees the best split is selected as follows.

1. The information acquired by choosing a feature can be computed through :

a) The entropy of Shannon, which measures the quantity of information

$$Entropy(p) = -\sum_{k=1}^{c} P(k/p) \times \log(P(k/p)) \tag{5.1}$$

where p is the number of examples associated to a position in the tree, c is the total number of classes, k/p denotes the fraction of examples associated to a position in the tree and labeled class k, $P(k/p)$ is the proportion of elements labeled class k at a position p.

Algorithm 1 Random forest algorithm

Input: Labeled training set S, Number of trees T, Number of features F.
Output: Learned random forest RF.

```
initialize RF as empty
for i in 1..T do
  S'_i ← bootstrap (S)
  initialize the root of tree i
  repeat
    if current node is terminal then
      affect a class
      go to the next unvisited node if any
    else
      select the best feature f* among F
      sub-tree ← split(S'_i, f*)
      add (leftChild, rightChild) to tree i
    end if
  until all nodes are visited
  add tree i to the forest
end for
```

b) The Gini index, which measures the dispersion in a population

$$Gini(x) = 1 - \sum_{k=1}^{c} P(k/p)^2 \tag{5.2}$$

where x is a random sample, c is the number of classes, k/p denotes the fraction of examples associated to a position in the tree and labeled class k, $P(k/p)$ is the proportion of elements labeled class k at a position p.

2. The best split is then chosen by computing the gain of information from growing the tree at given position, corresponding to each feature as follows :

$$Gain(p,t) = f(p) - \sum_{j=1}^{n} P_j \times f(p_j) \tag{5.3}$$

where p corresponds to the position in the tree, t denotes the test at branch n, P_j is the proportion of elements at position p and that go to position p_j, $f(p)$ corresponds to either $Entropy(p)$ or $Gini(p)$.
The feature that provides the higher Gain is selected to split the node.

The optimal training of a classification problem can be NP-complete. Tree ensembles have the advantage of running the algorithm from different starting points, and this can better approximate the near-optimal classifier.

In his paper, Leo Breiman discusses the accuracy of random Forests. In particular, he gave proof that the generalized error, although different from one application to another, always has an upper bound and so random forests converge [Breiman, 2001].

The injected randomness can improve accuracy if it minimizes correlation while maintaining strength. The tree ensembles investigated by Breiman use either randomly selected

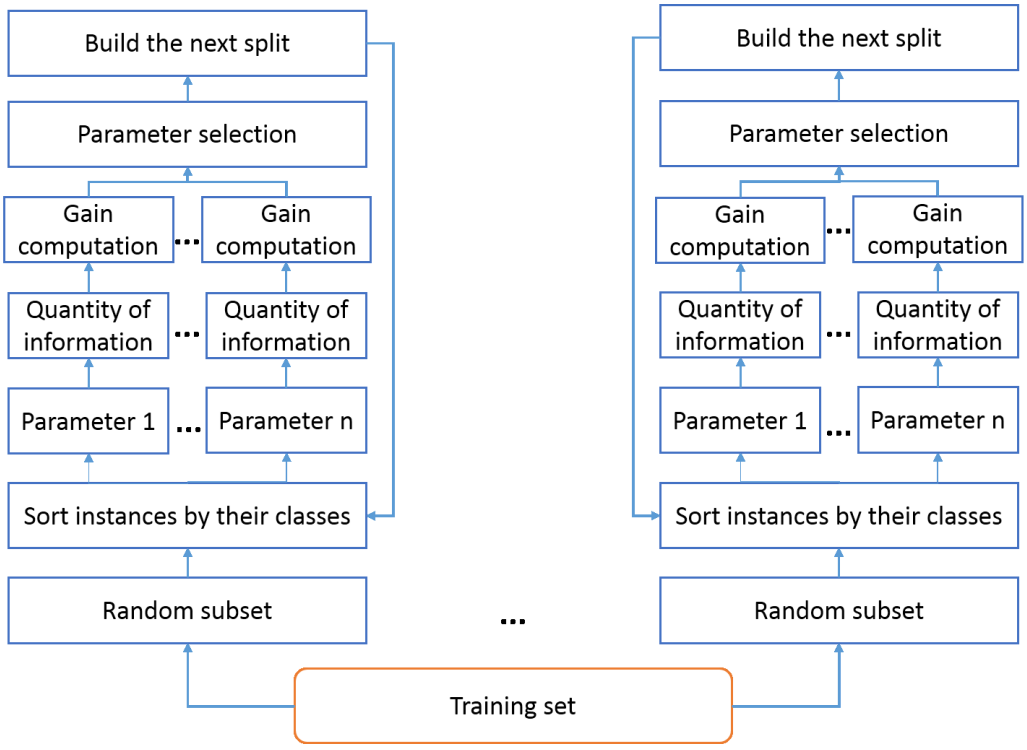


FIGURE 5.5 – Growth of a tree in the forest.

inputs or a combination of inputs at each node to grow the tree. These methods have interesting characteristics as :

– Their accuracy is at least as good as Adaboost.
– They are relatively robust to outliers and noise.
– They are faster than bagging or boosting.
– They give internal estimates of error, strength, correlation, and variable importance.
– They are simple and the trees can be grown in parallel.

There are four different levels of diversity which were defined in [Sharkey et al., 1997], level 1 being the best and level 4 the worst.

– **Level** 1 **:** no more than one classifier is wrong for each pattern.
– **Level** 2 **:** the majority voting is always correct.
– **Level** 3 **:** at least one classifier is correct for each pattern.
– **Level** 4 **:** all classifiers are wrong for some pattern.

RF can guarantee that at least level two is reached. In fact, a trained tree is only selected to contribute in the voting if it does better than random, i.e. the error rate generated by the corresponding tree has to be less than 0.5, or the tree will be dropped from the forest [Breiman, 2001].

[Verikas et al., 2011] argue that the most popular classifiers (Support Vector Machine SVM, Multi Layer Perceptron MLP, and Relevance Vector Machine RVM) provide too little insight about the variable importance to the derived algorithm. They compared each of these methodologies to the random forest algorithm to find that in most cases RF outperform other techniques by a large margin.

Data packet transfer consumes the highest amount of energy in the wireless sensor network. The higher the distance of transfer gets, the more energy is consumed. It is therefore preferable that the sensors communicate within the shortest radio range possible. Several solutions to preserve the network's energy have been investigated [Dam et al., 2003, YE et al., 2008], and they include the study of the topology. In this thesis, we compare several network topologies and study their impact on the quality of health assessment. This will allow us to choose an adequate topology for the data collected for health assessment.

5.4/ EVALUATION OF THE NETWORK TOPOLOGY

Data gathering in WSNs can be either periodic or event-driven. In periodic applications, data is gathered periodically while in event-driven applications gathering depends on the occurrence of some events. In both cases, the goal from aggregation is reducing energy dissipation by holding packets for as long as possible in intermediate nodes. All packets will be combined together then forwarded in the network. It is obvious to see that a decrease in energy consumption leads to an increase in the overall delay, and vice versa. A reliable solution would aim at finding an acceptable tradeoff between energy consumption and delay in WSNs [Kwon et al., 2011].

For periodic data gathering, data aggregation is achieved through organizing the network according to a logical structure ; mainly a tree [Solis et al., 2004] or a clustering [Toscano et al.,]. When a tree is used, aggregators are the internal nodes in the tree routed at the sink. With clustering structures, aggregators are the Cluster Heads (CH). In [Heinzelman et al., 2000, Jina et al., 2008], the authors prove that clustering methods provide better results for data aggregation, as they consume less energy. Defining a specific cluster and choosing the CH (aggregator node in the cluster) have an important impact on aggregation quality and energy consumption. Besides, structured approaches incur high maintenance overhead in event based applications. In fact, the source nodes change when a new event occurs. In other words, when the network starts running, the structure is fixed based on the positions of nodes sensing the event (source nodes). For the next round, the event may occur somewhere different in the network, which results in a change in source nodes. Consequently, the fixed structure will perform poorly [Yousefi et al., 2012].

Several WSN topologies were used in existing monitoring applications. [Kait et al., 2007] propose a WSN-based paddy growth monitoring system. Sensor nodes gather and send field data, such as temperature, periodically to the Base Station (BS). This is done by using multi-hop routing which is not considered energy efficient. Sensor nodes transmit data through the nearest neighbor which might lead to the longest path. Moreover, this routing protocol does not consider the energy level of the sensor nodes to generate transmission path. Another interesting study by Yoo et al. [Yoo et al., 2007] proposes a precision and intelligence agricultural system referred to as the Automated Agriculture System. The goal of this system is to monitor and control the growing process of melon and cabbage in a greenhouse. In the system, sensor nodes are organized in a parent-child tree structure. The nodes join the network by broadcasting a parent search packet. Furthermore, the nodes transmit data to the BS using three gateway nodes. However, the tree structure has a single point of failure. Yang et al. [Yang et al., 2007] developed an intensive WSN-based irrigation monitoring system. Sensor nodes are placed by this

system in widely separated clusters. Thus, sensor nodes consume much energy for transmitting data to remote nodes in other clusters. Chiti et al. [Chiti et al., 2005] propose next generation firm for Agro-food productions. This system uses Ambient Intelligence and WSNs. The proposed system provides feedback and adaptability to increase productions in Agro-food. However, the deployed WSN uses a dynamic flooding inefficient-energy routing protocol. This is due to the fact that a large number of messages are broadcasted. Village eScience for Life [Kabashi et al., 2008] is a WSN-based agriculture project. It is implemented in developing regions in Africa and uses dynamic zone-based topology. This project initially deploy sensor nodes into zones in such a way that each sensor node remains within the transmission range of the nodes of at least two zones and each node belonging to a zone elects nodes in neighboring zones to which it can connect with a minimum transceiver power. Hence, several graphs are generated and the graph requiring minimum transmission power is selected for routing. However, this routing protocol does not guarantee to eliminate sensing holes. COMMONSense Net (CNS) [Jacques et al., 2007] is another WSN-based agriculture monitoring project developed for semiarid regions in developing countries. The routing protocol of CNS uses tree structure which is not reliable since a link failure or sensor node failure can make other nodes unreachable to BS. Unlike the earlier works that focus mainly on the WSN-based monitoring applications, recent research [Chavez et al., 2006] has significantly considered studying the actual structure of WSN through graph theory. In particular, geometric graphs are used in WSNs [Ke et al., 2009] to model the relationship between a sensor node and its neighboring sensor nodes [Gabriel et al., 1969, Matula et al., 1980].

5.5/ THE INVESTIGATED TOPOLOGIES

In order to illustrate the impact of WSN topologies on the quality of health estimations, we consider 90 sensor nodes ; 30 nodes for each of the monitoring parameters : temperature, Pressure, and Humidity. The sensors are randomly placed in the simulation window, and are equipped with batteries of $100j$. The sink is also placed randomly. With every data transfer, the energy of a sender is reduced regarding its distance from the recipient.

- Under normal conditions, temperature sensors follow a Gaussian law of parameter $(20\times (1+0.005t), 1)$, while these parameters are mapped to $(35, 1)$ in case of a malfunction of the industrial device. These sensors return the value 0 when they break down.
- When both the industrial device and the pressure sensors are in normal conditions, the value of pressure is computed as $(x \div 2 + 10)$, where x is the value of temperature. The parameters are changed to $(15, 1)$ in case of industrial failure, while the pressure sensors return 1 when they are themselves broken down.
- For a well-functioning device, the 10 humidity sensors produce data in the form of $(x \times 525 + 42)$. These parameters are set to $(70, 10)$ in case of device failure, while malfunctioning humidity sensors produce the value 0.

The probability that a failure occurs at time t follows an exponential distribution of parameter $1 \div 100$. Data is generated as follows.

for each time unit $t = 1..200$ during the industrial device monitoring **do**

  for each category c (temperature, pressure, humidity) of sensors **do**

for For each sensor s belonging to category c **do**

if s has not yet detected a device failure **then**
s picks a new data, according to the Gaussian law corresponding to a well-functioning device, which depends on both t and c
a random draw from the exponential law detailed previously is realized, to determine if a breakdown occurs on the location where s is placed
else
s picks a new datum according to the Bernoulli distribution of a category c sensor observing a malfunctioning device
end if
end for
end for
end for

The global failure level F^t of a set of 90 sensed data produced by the wireless sensor network at a given time t is defined as follows :
For each sensed datum $d_i^t, i = 1..90$, let $f_i^t \in \{1, .., 5\}$ be the functioning level related to its category (pressure, temperature, or humidity). Then $F^t = \max f_i^t \mid i = 1..90$.

Values of parameters	T : $(1 + 0.005t), 1$
	P : $((1 + 0.005t), 1) \div 2 + 10$
	H : $((1 + 0.005t), 1) \times 525 + 10$
Failure values	T : (35,1)
	P : (15,1)
	H : (70,10)
Number of regular sensors	90
Energy of a regular sensor	100 j
Number of CHs	16
Energy of a CH	1000j
radio range	0.2 m
Sink coordinates	(1,1)
Failure probability	$1 \div 100$
Simulation time limit	200 units

TABLE 5.1 – Simulation characteristics.

In the following, we propose 2 different topologies and 2 different scenarios for each.

5.6/ SIMULATION RESULTS

5.6.1/ NETWORK CONNECTIVITY

In order to illustrate the impact of topologies on the quality of data at the sink level, we consider 200 sensor nodes randomly placed in a surveillance area, and equipped with batteries of $100\,j$. The sink, by default, is located at the top left corner of the simulation window. In the following, we propose four different topologies.

5.6.1.1/ SCENARIO 1

In this first scenario, we consider that the nodes are grouped into 16 clusters. Each cluster is managed by a leader called Cluster Head (CH). Each CH has $1000\,j$ for battery supply and a radio range of 0.2 m, each sensor node within this range can communicate with the respective CH. If a sensor node happens to be in the radio range of more than one CH, it communicates only with the closest one. When a sensor captures new information (within a radio range of 0.2 m), it sends it to its CH. The latter aggregates the received data packets into one packet and then forwards it towards the sink. The resulting topology is shown in Figure 5.6(a).

The sink is at the top left corner of the window, the aggregators are colored in red and regular nodes in blue. At the starting time $t = 0$, each node in the network is connected to the node it communicates with. The link between a regular node and an aggregator is illustrated by a black edge, and the link between the aggregators and the sink by a green one (see Figure 5.6(a)).

In figure 5.6(b), we can see that the nodes that exhaust their energy supply first are the CHs that are located the furthest from the sink. This is due to the long distance of packet transfer. Consequently, the nodes in the corresponding cluster loose their connections. In round 1 alone, two CHs are down and therefore two regions are no longer monitored.

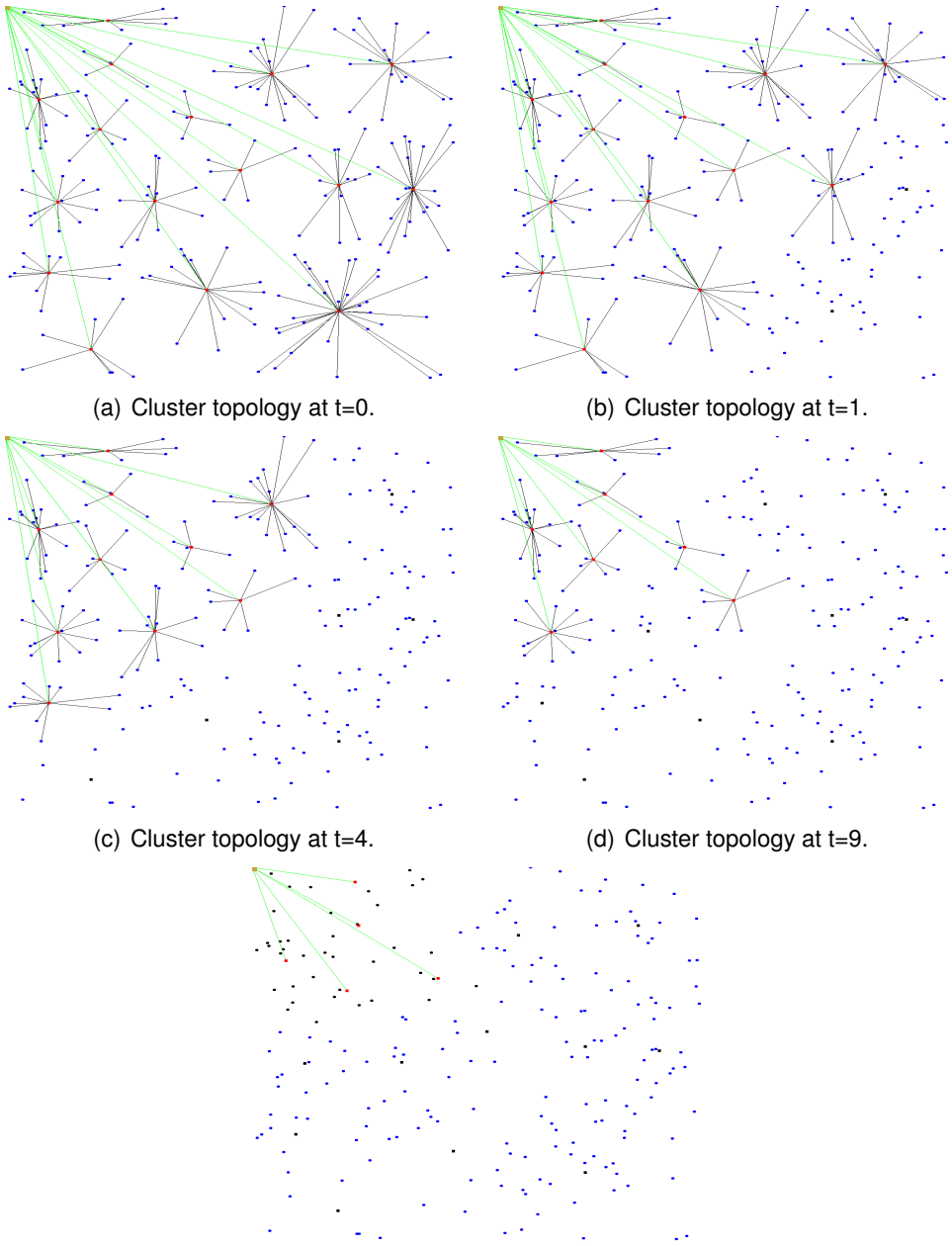
Figure 5.6(c) shows the evolution of the topology at $t = 4$. Through the simulation, the aggregators that are the furthest from the sink continue to exhaust their batteries (notice their color changing to black) and therefore all the edges connecting them to other nodes are broken. As time evolves, more sensor nodes consume all their disposable energy. The further a node is from the sink, the sooner it is dropped from the network due to long distance of transfer (Figure 5.6(d)).

At the end of the simulation ($t = 10$), and as shown in Figure 5.6(e), only the nodes within a short distance from the sink node managed to keep enough energy to communicate with the sink. Therefore, the events taking place in further areas of the network can neither be detected nor reported. Nevertheless, this missing information can be relevant to health assessment. The simulation ends when no new events can be reported to the sink.

5.6.1.2/ SCENARIO 2

We suggest, in this scenario, that all sensor nodes in the network have the same role and importance ; i.e. there is no aggregation role, no clusters, and no CHs. Data packets are forwarded in a hop-by-hop manner like described in the following and shown in Figure 5.7(a). Each sensor is able to discover its neighbors within a radio range of 0.2 m. We assume that every node can access information about its neighbors, including their locations. Therefore, a sensor node is able to choose an other node to communicate with. The latter has to be the closest node to the sink within the sender's radio range. In other words, every sensor is most likely to communicate with the furthest neighbor within its radio range.

Sensor nodes are organized in a tree hierarchy from the sink (being the root of the tree), until sensor nodes having no descendants (leaf nodes). In a tree topology, all sensor nodes have the same role ; i.e. they all participate at discovering the neighborhood and building the topology. On the contrary, in a cluster topology, CHs are given an extra energy



(a) Cluster topology at t=0.

(b) Cluster topology at t=1.

(c) Cluster topology at t=4.

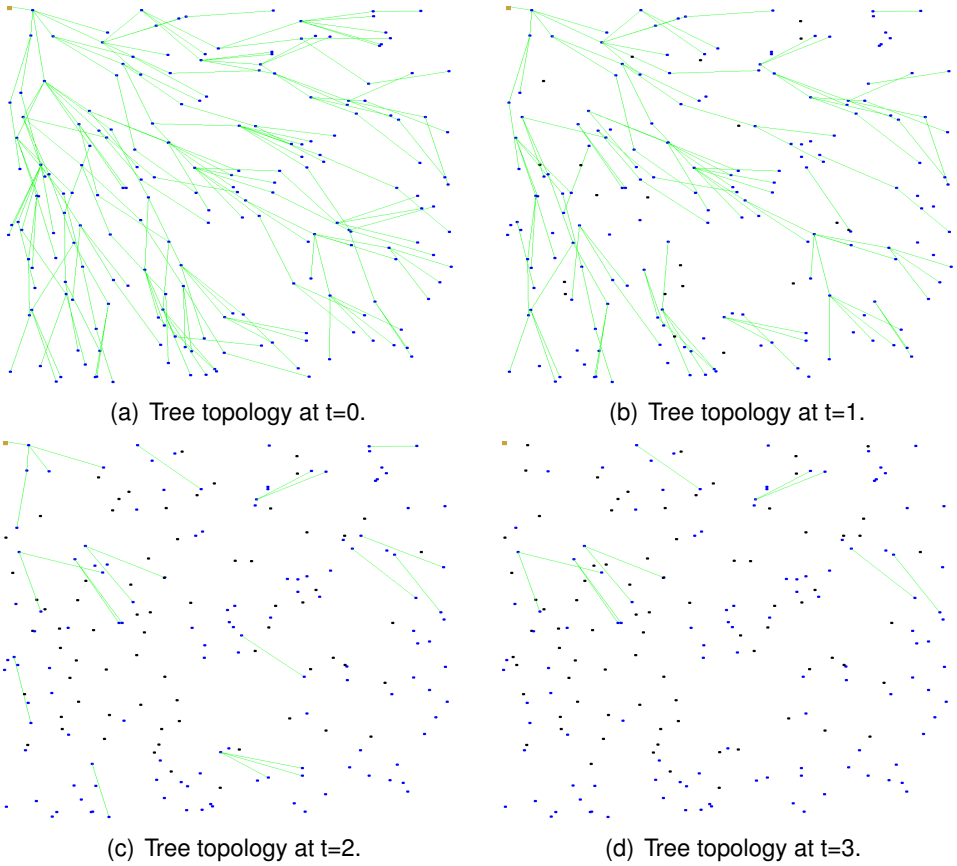
(d) Cluster topology at t=9.

(e) Cluster topology at t=10.

FIGURE 5.6 – Cluster topology.

supply and are the nodes responsible for building the links between sensor nodes towards the sink. Therefore, the batteries last longer and we dispose with more data for health assessment.

As soon as the simulation starts (Figure 5.7(b)), the nodes transferring data packets for the longest distances start to fail. At $t = 2$, only the data sensed in the area around the sink is transferred. Few other connections between nodes still remain in the network, but with no link to the base station, as shown in Figure 5.7(c). At $t = 3$, the simulation has already stopped due to the disappearance of every connection with the sink (Figure 5.7(d)).



(a) Tree topology at t=0.

(b) Tree topology at t=1.

(c) Tree topology at t=2.

(d) Tree topology at t=3.

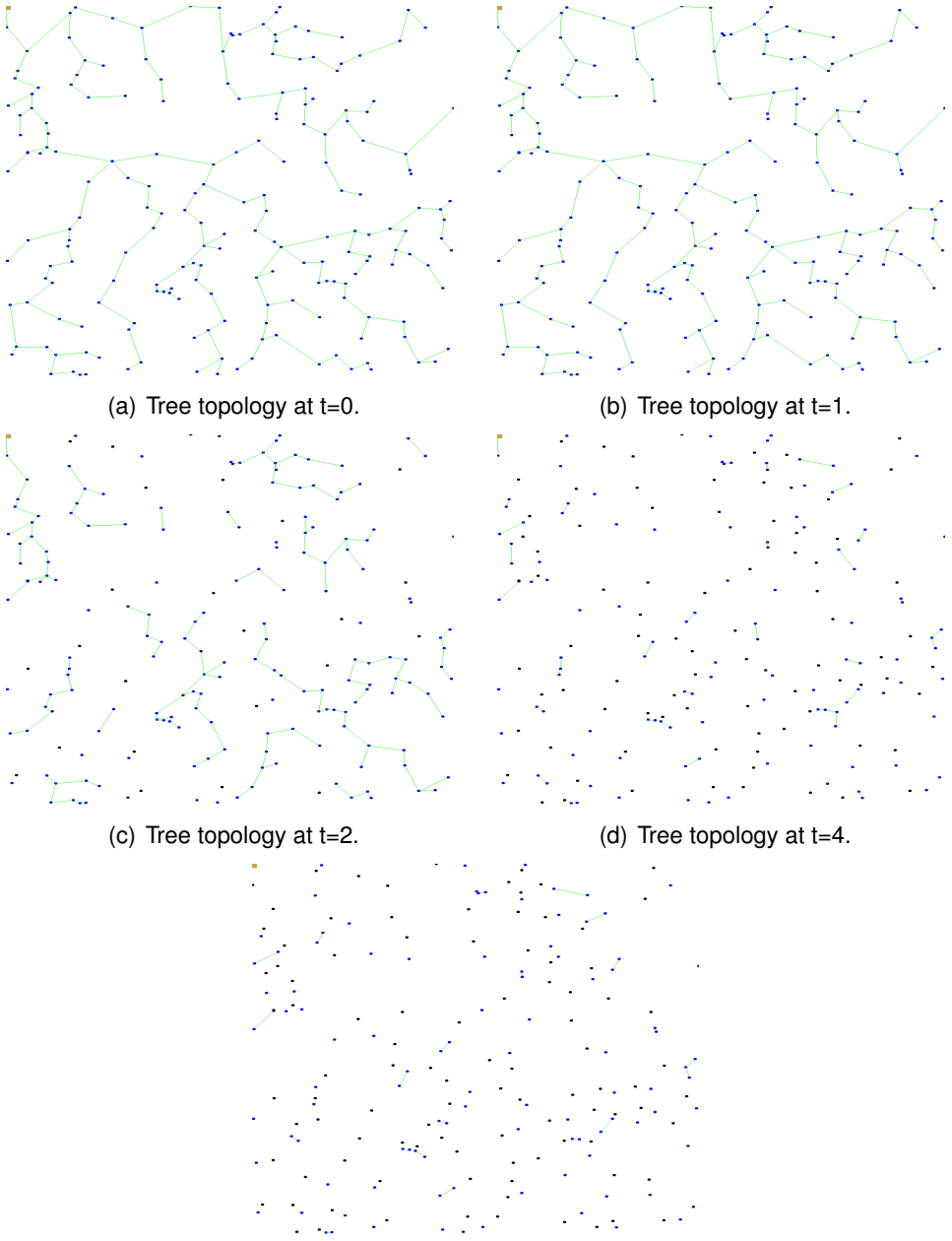
FIGURE 5.7 – Tree topology.

Comparing to the previous topology, the simulation time is 3 times shorter. This means that the energy consumption is 3 times higher in this tree topology comparing to the previously simulated cluster topology. Organizing the network in clusters seems to help reduce the number of total transmissions due to data aggregation by the CH. But so far, we can not tell for sure if this is due to the network topology, or to the distance of transfer itself. In Section 5.6.1.3, we investigate this further by changing the distance of transfer in the proposed tree topology.

5.6.1.3/ SCENARIO 3

Our hypothesis is that an important amount of energy in the network is consumed due to packet transfer. In order to highlight the impact of data transfer on energy consumption, we adapt the same topology from Section 5.6.1.2. This time, each node communicates with an other node that is within its radio range and that guarantees the shorter distance of transfer while getting closer to the sink. This means that every node would most likely communicate with the closest sensor within its radio range, which is almost the opposite of the previous scenario. This topology is illustrated in Figure 5.8(a).

At the end of the first round of data collection (Figure 5.8(b)), most of the links between the sensors are still holding (apart from one or two links). We only start to see a clear network disconnection in Figure 5.8(c), where only a small portion of the network is still



(a) Tree topology at t=0.

(b) Tree topology at t=1.

(c) Tree topology at t=2.

(d) Tree topology at t=4.

(e) Tree topology at t=5.

FIGURE 5.8 – Optimized tree topology.

connected with the sink. At $t = 4$, only one sensor node seems to be able to deliver data to the sink. The end of this topology's simulation is illustrated in Figure 5.8(e). As we can see, the sink node can no longer receive information from any node in the network. It is therefore no longer impossible to perform data processing at the sink level. Even though some nodes still have enough energy to detect the occurrence of new events, they have lost the connections they had with other nodes helping deliver information towards the sink node.

Tuning the network parameters in a way that reduces the distance of transfer, only helped us gain two extra rounds of data collection. This is in comparison with scenario described

in Section 5.6.1.2. Even with this slight improvement, the cluster topology still provided a longer network lifetime. Clearly, the deployment of the tree topology is not that beneficial for good health assessment.

5.6.1.4/ SCENARIO 4

Here we consider the same topology as described in Section 5.6.1.1. In order to reduce the amount of energy spent on packet transfer distance, we suggest that the CHs do not send the information directly to the sink. Instead, each CH forwards its data packets to the closest neighbor CH within its radio range. This CH is only chosen if it is geographically closer to the sink. In other words, the data collected in each cluster is forwarded in a hop-by-hop manner towards the sink via the CHs. This can be described as a tree of clusters. This topology is illustrated in Figure 5.9(a).

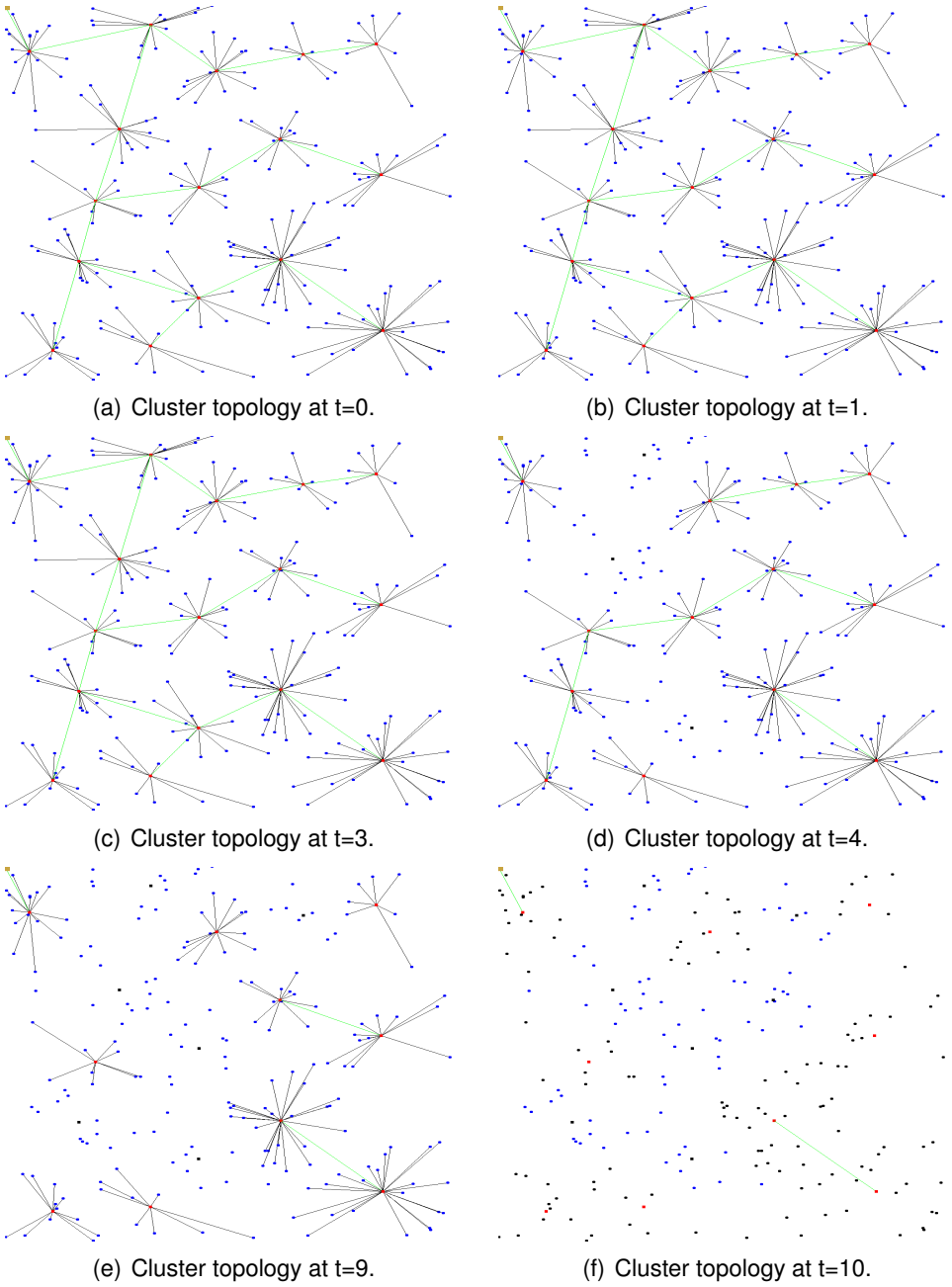
At $t = 1$, none of the sensor nodes has exhausted its energy supply yet. Of all the simulated topologies, this is the first one that shows a full connectivity after the first round of data collection (Figure 5.9(b)). At $t = 3$, all the nodes are still able to deliver the detected events towards the sink. In Figure 5.9(d), we witness the first signs of network disconnection. For the same number of rounds, in the previous cluster topology, half the network was already disconnected (see Figure 5.6(c)). This configuration does not improve the network's lifetime when compared to the results provided in Section 5.6.1.1. Nevertheless, due to a better network connectivity, more data packets are delivered towards the sink. Consequently, the quality of health assessment can be improved with a better quality data.

As a conclusion, the distance of transfer has a clear impact on the energy consumption, and therefore the nodes' lifetime. The more the nodes live, the longer the network's connectivity is maintained. Data aggregation reduces the number of packet transfer, and therefore further reduces the overall energy consumption in the network.

In the light of these results, we collected data for health assessment by connecting the nodes in a clustering topology. The description of this topology and the obtained health assessment results are presented herafter.

5.6.2/ HEALTH ASSESSMENT

In the first scenario, when a node senses new data, it forwards it directly to the BS. At the end of each round, the sink will receive 30 different measures of temperature, pressure, and humidity each. The sink will only keep one value of each parameter. This is guaranteed by computing an average using a Gaussian distribution.
In the second scenario, 9 sensors are added to the topology. These sensors will be the aggregators (3 per parameter). Therefore, the topology now presents 9 clusters and in each, nodes send the sensed data to the CH. The CH aggregates the data packets from each round and sends the computed value of the relative parameter to the sink node. It should be noted that at this step, the CHs are placed randomly and their distance to their cluster members is not optimized.
In the third and last topology, we also considered 9 clusters. This time after all the sensors (CHs and regular nodes) are placed, each regular node finds the closest CH to it and adapts the same type (i.e. parameter).



(a) Cluster topology at t=0.

(b) Cluster topology at t=1.

(c) Cluster topology at t=3.

(d) Cluster topology at t=4.

(e) Cluster topology at t=9.

(f) Cluster topology at t=10.

FIGURE 5.9 – Optimized cluster topology.

We collect data in the network using the algorithm described in Section 5.5. After data collection step, health assessment is performed through the RF algorithm described in Section **??**. Nodes that capture new data packets forward the information (according to the corresponding network topology) towards the sink for processing. The data is then fed to the RF algorithm to assess the health of the monitored device.

We varied the number of trees in the forest from 1 to 100, and obtained in total 18 different forests. For each forest, we repeated the simulation 10 times. During the simulation, the sensors communicate the data generated following the laws described in Section 5.5. The simulations are timed, i.e. the simulation does not end when the system fails, but

when the simulation time is reached. The decision for each tree is averaged over the 10 simulations, and the final decision is averaged over all the decisions given by each tree in the forest. In the following, we show the average number of errors in health estimation for each of the 3 proposed topologies.

In Figure 5.10 we plotted the average number of errors in health estimation, when all nodes can communicate with the BS. The error rate was maintained below 50 $\%$ at all times. With the number of trees increasing in the forest, the error rate decreases and gets close to 0 $\%$. When the number of trees in the forest is more than 9, the error rate becomes almost constant.

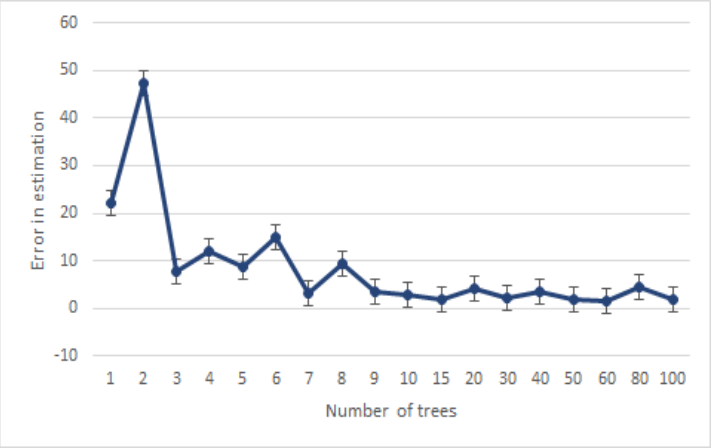


FIGURE 5.10 – Error in health estimation for the star topology.

Figure 5.11 shows the average number of errors in health estimation, when data is aggregated before being sent to the BS (as described in Section 5.5). The error rate, compared to the previous simulation, was reduced by half, and was stabilized when number of trees is greater than 20. Aggregating data reduces the frequency of transferring packets in the network ; CHs will receive data from nodes within their range, combine them together and send them as one packet. As a result, the overall activity of sensors will be reduced, and consequently they will consume less energy. This means that sensors can live longer (comparing to the previous topology) to ensure transferring relevant data to the BS for health assessment. We can therefore conclude that reducing the number of packets in the network helps improve the quality of health assessment.

In Figure 5.12 we plotted the average number of errors in health estimation, when nodes forward their data to the nearest aggregator. Error rate was reduced by almost a half when the distance of transfer is reduced, and reached 0 $\%$ when the number of trees is greater than 80. Transferring data over a short distance requires less energy from the sender. This helps preserve energy for a longer period and ensures that data needed for health assessment can be delivered to the BS over that time period.

To summarize, aggregating data packets ensures that nodes degrade gracefully (rather than abruptly) and results in more accurate estimations. Also, having nodes transfer their data over a short distance also helps preserve the available energy in the network. The point from which the error rate is stabilized can be considered as the optimal (or minimum) number of trees needed in the forest.

The training set is obtained by simulating 100 observations for 10 successive times, which results in 1000 instances. The resulting data base is then used to train 100 trees that will

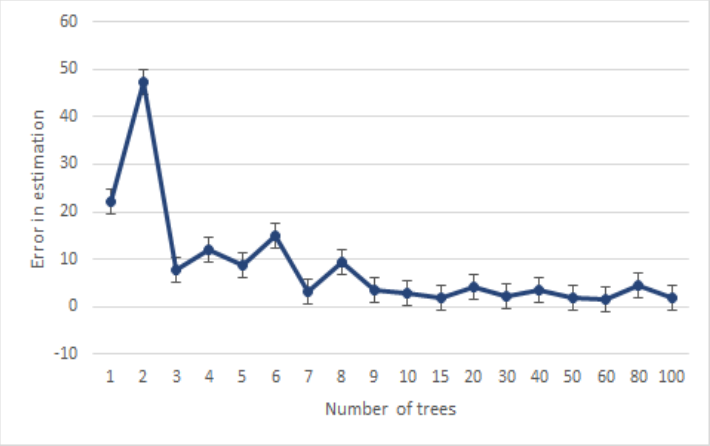


FIGURE 5.11 – Error in health estimation for cluster topology.

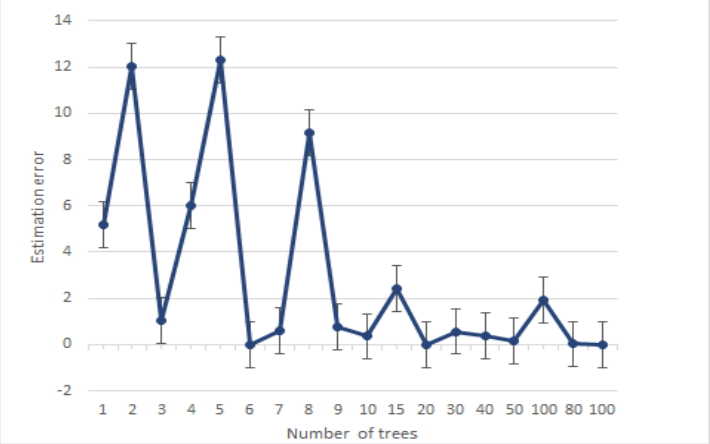


FIGURE 5.12 – Error in health estimation for cluster topology with closest aggregator.

constitute the trained random forest.

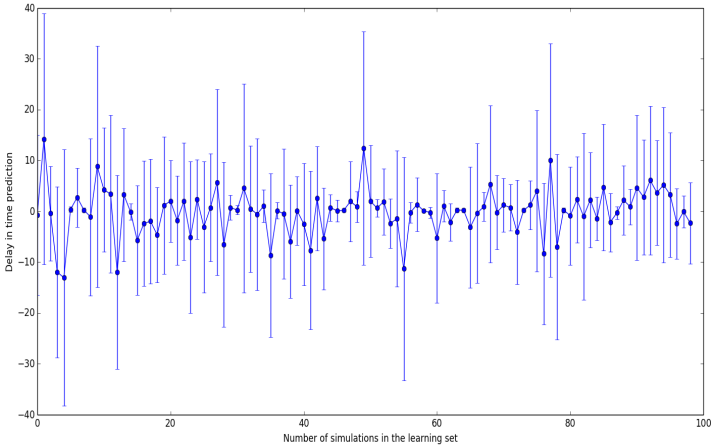


FIGURE 5.13 – Delay in failure detection with respect to the number of simulations.

Figure 5.13 presents the delay between the time the system enters a failure mode and

the time of its detection. This is done in the absence of correlations between the different features. The 0 time value of delay, the negative values, and positive value refer to in-time predictions, early predictions and late predictions of failures, respectively. The plotted values are the average result per number of simulations which varies from 1 to 100. With time, sensor nodes start to fail in order to simulate missing data packets. As a result, the RF algorithm was able to detect 54 % of the failures either in time or before their occurrence.

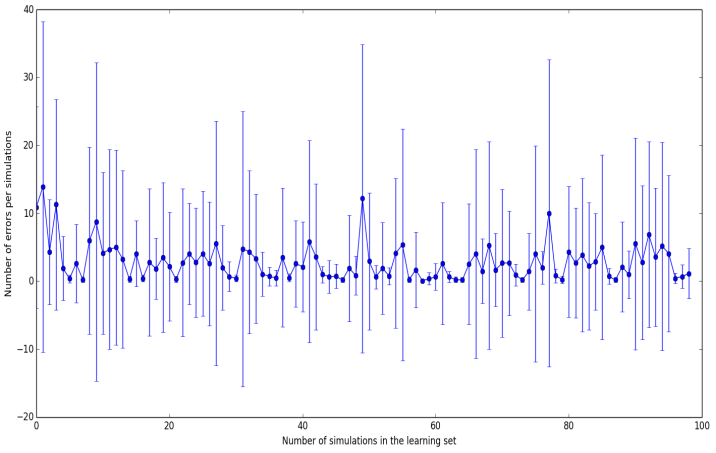


FIGURE 5.14 – Error rate in health assessment with respect to the number of simulations.

For each of the 100 performed simulations, we calculated the average number of errors in fault detection, produced by the trees in the forest. Figure 5.14 shows that this error rate remained below 15 % through the simulation. This error rate includes both "too early" and "too late" detections. When certain sensor nodes stop functioning, this leads to a lack on information, which has an impact on the quality of predictions ; this explains a sudden increase in the error rate with time. We can conclude from the low error rate in the absence of some data packets that increasing the number of trees in the RF helps improve the quality and accuracy of predictions.

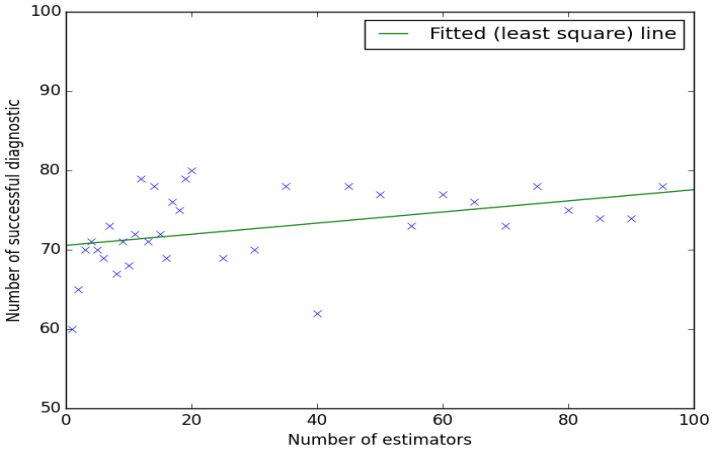


FIGURE 5.15 – Number of successful health assessments with respect to the number of trees.

As described in Section 5.5, a correlation was introduced between the features. Figure 5.15 shows the number of successful fault detection when the number of tree estimators

in the forest changes. As shown in this figure, the RF method guarantees a 60 % success rate when the number of trees is limited to 5. As this number grows, the accuracy of the method increases to reach 80 % when the number of trees is around 100. Comparing to the previous results, the correlation between the features helps decrease the uncertainties in health assessment when the number of trees increases. The algorithm is able to understand the relationship between two features. Thus, when some values describing a feature are missing, the algorithm can deduct them from the available information about the rest features.

5.7/ CONCLUSION

The random forest algorithm was used for health assessment, when the industrial device is monitored by a wireless sensor network. Data in a wireless communication can easily be lost, and the proposed algorithm was able to quickly adapt to the situations where the provided information was incomplete. This was illustrated by the peaks in the error rate plots once after few simulations.

The impact of network topology was also investigated. We compared the star topology to the cluster topology, by looking into the quantity of data delivered to the sink when each of the topologies is employed. We concluded that organizing the network into clusters helps preserve the network's energy, and therefore improve the quality of data used in the health assessment process. While employing the cluster topology, we varied the distance of transfer by varying the distance of communication between the sensor nodes. The simulation results proved that a short packet transfer distance helps further preserve the overall energy in the network.

III

CONCLUSION

6

CONCLUSION AND FUTURE WORK

Prognostics and health management (PHM) has become a very important tool in modern industry. It is the core activity of condition-based maintenance. PHM helps increase availability, reliability, and safety by predicting the remaining useful life of engineering assets. This activity aims at planning a maintenance before the system faces a failure, and thus a shutdown can be avoided. Many approaches were reported in the literature, and among all models, data-driven ones were most used. These models have the advantage being built faster and at a lower cost comparing to the other approaches.

As their name indicates, data-driven models are built upon data. Relevant parameters are determined and monitored. Their values, crossed with the corresponding state, will be used to build a degradation model for the system under consideration. This monitoring activity is usually performed by means of independent sensors. For complex systems (ones that have different types of parameters to monitor) or systems that spread on a large area (monitoring needs to be done in different spots), the sensors are most likely to be connected through physical wires. For some applications, the use of wired networks gives rise to a number of drawbacks such as high costs, difficulty of access, single point of failure, etc. Wireless sensor networks (WSN), with their low cost, fault tolerance, and easy deployment are a solution for these problems.

Unfortunately, the problems cannot be solved quite easily. WSN are vulnerable, prone to failures, and data packets are very likely to be lost in the transfer process. Before deployment, some techniques need to be put in place to ensure the reliability of the network and reduce the amount of the lost information.

In this thesis, we proposed a distributed algorithm for resiliency in WSN. The algorithm ensures that in a surveillance area (defined by a distance equal to the sensors radio range), one and only one sensor performs the surveillance activity. The other nodes, that are not needed for the monitoring activity are put to sleep. This helps reduce the overall energy consumption by preserving the energy of the sensors not participating in the monitoring activity. In the purpose of limiting data loss, an active node will copy the data it senses onto its neighbors. Once the active node fails, the new elected node will retrieve a copy of the lost data and send it again.

Prognostics and health management approaches are based on the assumption of completeness of data. In this thesis, we challenged this belief and placed ourselves in the case where the monitoring data is incomplete. For the health assessment step, we proposed the use of the random forest algorithm. We injected random errors in the network, which forced the sensors to randomly fail. We proved that the algorithm is able to adapt to the changes in the quality of the monitoring conditions.

To further test the algorithm, and illustrate the impact of the data gathering mechanism on the quality of data, we varied the network topologies and compared the results of health assessment.

We conducted simulations of each of the proposed solutions and discussed their results. In the following we enumerate the contributions of this thesis regarding the identified challenges.

1. To the best of our knowledge, existent research works in the prognostics and health management field used either individual sensors, or sensors that are connected with physical wires for the monitoring activity. We challenged the feasibility of this, as the use of wireless sensor networks is sometimes a requirement rather than a choice. Consequently, in this thesis we addressed some issues and challenges that have not been questioned before, mainly the completeness of data used in the prognostics process. In the highlight of this, we proposed a new condition-based maintenance flowchart, one that includes the steps to go through before the monitoring device is ready for deployment.
2. The more information we dispose of, the more accurate the estimations are. Unfortunately, wireless communications are by nature prune to failures. A simple route update, or a packet interference can cause an information permanent loss. Solutions for data reliability were reported in the literature. Most of them relied on retransmission and redundancy mechanisms. We argued that these two mechanisms do not help preserve the disposable energy, which imposed one of the highest limitations to wireless sensor networks. Therefore, we proposed a distributed algorithm that insures resiliency in sensor networks by recovering lost data packets. This algorithm has the particularity of avoiding any unnecessary packet transfer ; an information is retrieved only when needed and without employing an acknowledgment mechanism.
3. To further ameliorate the quality of health assessment, we looked at data loss from another perspective. Reliability mechanisms are unable to guarantee to total elimination of data loss. For this, we gained benefit from the particularity of the random forest algorithm. The random distribution aspect, gives the algorithm more precision and a useful variation thanks to its different starting points. Since the trees in the forest are different regarding the initial distribution, the algorithm was able to quickly recover from missing data packet at the processing step. We were able to successfully assess the system's state of health with an injected error in the network. this error aimed at forcing the nodes to fail and therefore reduce the amount of information at the processing unit.

6.1/ LIMITATIONS

The dependability of wireless sensor networks has not been fully studied in this research work. Studying the reliability of the network, although improves the completeness of data, does not ensure its correctness. Securing the network will play a role in improving the quality of health assessment.

The random forest algorithm is sensitive to the number of parameters taken into consideration. While a small number features would not be sufficient for reliable estimations,

numerous features will lead to extensive computations, leaving a portion of them out of use. Finding the adequate number can be a difficult task in some applications. This method also still needs to be validated once the results are incorporated in the remaining useful life estimation.

Although the performed simulations gave good results, it would be interesting to test the developed algorithms on a real application and verify the accuracy of the results in that case.

6.2/ FUTURE WORK

Prognostics and health management is a challenging area of study. Considering the progress that has been achieved, and on the basis of the work accomplished here, we address some challenges to improve the quality of health assessment, diagnostics, prognostics, and decision making.

1. Most of the existing research work is limited to the use of condition monitoring (CM) data. Event data can be kept record of and combined with CM data to increase the accuracy of health assessment, diagnostics, prognostics, and decision making. CM data only provides quantitative information about specific monitored parameters. On the other hand, event data provides indications about maintenance activities, repair actions, system breaking down, etc. Gaining knowledge about all events as they take place helps the model better represent the behavior of the system (or component) and its degradation model.
2. Training is mostly done off-line, and in some cases the time dedicated to data collection could be limited. In other cases, industrials withhold some information due to confidentiality issues. But both situations mean that the data collected for the training set does not necessarily reflect all the possible scenarios that could possibly be encountered. For this reason, it would be interesting to update the model online in a way that rectifies the gap between the predicted class/RUL and the observed class/RUL.
3. Wireless networks are easily hacked, and this is one of the reasons why they are not used for industrial monitoring. Implementing a security code will protect the network from attacks. Consequently, data will be protected from alterations, misrouting, and falsification. Therefore, at the base station, data would be more accurate, and thereby the predictions also.
4. The proposed algorithm showed good results for health assessment. The first results are encouraging to take the research further by either using the same algorithm for diagnostics and RUL prediction, or using a different algorithm on the basis of the results that are the outputs of the RF algorithm at the health assessment step.

BIBLIOGRAPHIE

[Akan et al., 2005] Akan, O. B., et Akyildiz, I. F. (2005). **Event-to-sink reliable transport in wireless sensor networks**. *IEEE/ACM Transactions on Networking*, 13(5) :1003–1016.

[Akyildiz et al., 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., et Cayirci, E. (2002). **A survey on sensor networks**. *IEEE Communications Magazine*, 40(8) :102–114.

[Al-Wakeel et al., 2007] Al-Wakeel, S. S., et Al-Swailem, S. A. (2007). **Prsa : A path redundancy based security algorithm for wireless sensor networks**. Dans *IEEE Wireless Communications and Networking Conference, WCNC*, pages 4159–4163, Hong Kong, China.

[Avizienis et al., 2000] Avizienis, A., Lapire, J.-C., et Randell, B. (2000). **Fundamental concepts of dependability**. Rapport technique, University of Newcastle.

[Bae et al., 2014] Bae, S., Cha, H., et Suh, Y. (2014). **Study on condition based maintenance using on-line monitoring and prognostics suitable to a research reactor**. Dans *Second European Conference of the Prognostics and Health Management Society*, pages 810–814, Nantes, France.

[Bahi et al., 2011] Bahi, J., Haddad, M., Hakem, M., et Kheddouci, H. (2011). **Distributed lifetime optimization in wireless sensor netwoks**. Dans *HPCC*, pages 432–439.

[Bahi et al.,] Bahi, J. M., Guyeux, C., et Makhoul, A. **A security framework for wireless sensor networks : Theory and practice**.

[Bahi et al., 2013] Bahi, J. M., Haddad, M., Hakem, M., et Kheddouci, H. (2013). **Stabilization and lifetime optimization in distributed sensor networks**. Dans *AINA*, pages 437–442.

[Baruah et al., 2005] Baruah, P., et Chinnam, R. (2005). **Hmms for diagnostics and prognostics in machining process**. *International journal of Production Research*, 43(6) :1275–1293.

[Batko, 1984] Batko, W. (1984). **Prediction Method in Technical Diagnostics**. PhD thesis, Cracov Mining Academy.

[Benenson et al., 2005] Benenson, Z., Gedicke, N., et Ravivo, O. (2005). **Realizing robust user authentification in sensor netwoks**. Dans *Real-World Wireless Sensor Networks (REALWSN'05)*.

[Braginsky et al., 2002] Braginsky, D., et Estrin, D. (2002). **Rumor routing algorithm for sensor netwoks**. Dans *1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 22–31, NY, USA. ACM Press.

[Breiman, 1996] Breiman, L. (1996). **Bagging predictors**. *Machine Learning*, 24 :123–140.

[Breiman, 2001] Breiman, L. (2001). **Random forests**. *Machine Learning*, 45 :5–32.

[Brotherton, 2000] Brotherton, T. (2000). **A testbed for data fusion for engine diagnostics and prognostics**. Dans *IEEE Aerospace Conference*, pages 163–171, Big Sky MT.

[Bunks et al., 2000] Bunks, C., McCarthy, D., et Al-Ani, T. (2000). **Condition-based maintenance of machines using hidden markov models**. *Mechanical Systems and Signal Processing*, 14(4) :597–612.

[Cadini et al., 2009] Cadini, F., Zio, E., et Avram, D. (2009). **Model-based monte carlo state estimation for condition-based component replacement**. *Reliability Engieneering and System Safety*, 94 :752–758.

[Carman et al., 2000] Carman, D. W., Kuus, P. S., et Matt, B. J. (2000). **Constraints and approaches for distributed sensor network security**. Rapport technique, NAI Labs, The Security Research Division, Network Associates, Inc. Glenwood.

[C.Cempel, 1987] C.Cempel (1987). **Simple condition forecasting techniques in vibroacoustical diagnostics**. *Mechanical Systems and Signal Processing*, 1 :75–82.

[Chavez et al., 2006] Chavez, E., Dobrev, S., Kranakis, E., Opatrny, J., Stacho, L., Tejeda, H., J., et Urrutia (2006). **Half-space proximal : a new local test for extracting a bounded dilation spanner**. Dans *the International Conference On Principles of Distributed Systems*, page 235âĂŞ245, Pisa, Italy.

[Chen et al., 2009] Chen, X., Kim, Y.-A., Wei, W., Shi, Z. J., et Song, Y. (2009). **Fault-tolerant monitor placement for out-of-band wireless sensor network monitoring**. *Journal of Information Science and Engineering*, 25 :237–287.

[Cheng et al., 2013] Cheng, J., Ye, Q., Jiang, H., Wang, D., et Wang, C. (2013). **Stcdg : An efficient data gathering algorithm based on matrix completion for wireless sensor networks**. *Wireless Communications, IEEE Transactions on*, 12(2) :850–861.

[Chiti et al., 2005] Chiti, F., Cristofaro, A. D., Fantacci, R., Tarchi, D., Collodo, G., Giorgett, G., et Manes, A. (2005). **Energy efficient routing algorithms for application to agro-food wireless sensor networks**. Dans *the IEEE International Conference on Communication (ICC)*, page 3063âĂŞ3067, Seoul, Korea.

[Choi et al., 2009] Choi, J., Hahn, J., et Ha, R. (2009). **A fault-tolerant adaptive node scheduling scheme for wireless sensor networks**. *Journal of Information Science and Engineering*, 25 :237–287.

[Crevecoeur, 1993] Crevecoeur, G. (1993). **A model for the integrity assessment of ageing repairable systems**. *IEEE Transactions on Reliability*, 42(1) :148–155.

[Dam et al., 2003] Dam, T. V., et Langendoen, K. (2003). **An adaptive energy-efficient mac protocol for wireless sensor networks**. Dans *Proceedings of the 1st international conference on Embedded networked sensor systems*, New York, USA.

[de Souza et al., 2007] de Souza, L. M. S., Vogt, H., et Beigl, M. (2007). **A survey on fault tolerance in wireless sensor networks**. Rapport technique, Computer Science University of Karlsruhe, Germany.

[Deng et al., 2004] Deng, J., Han, R., et Mishra, S. (2004). **Countermeasures against traffic analysis attacks in wireless sensor netwoks**. Rapport technique, University of Colorado.

[Dietterich, 2000] Dietterich, T. G. (2000). **An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization**. *Machine Learning*, 40 :139–157.

[Dijkstra, 1974] Dijkstra, E. W. (1974). **Self-stabilizing systems in spite of distributed control**. *Communications of the ACM*, 17(11) :643–644.

[Dong et al., 2007a] Dong, M., et He, D. (2007a). **Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis**. *European Journal of Operational Research*, 178 :858–878.

[Dong et al., 2007b] Dong, M., et He, D. (2007b). **A segmental hidden semi-markov model (hsmm)-based diagnostics and prognostics framework and methodology**. *Mechanical Systems and Signal Processing*, 21 :2248–2266.

[Douceur, 2002] Douceur, J. R. (2002). **The sybil attack**. Dans *Proceedings of the first International Workshop on peer-to-peer systems (IPTPS'02)*.

[Duane, 1964] Duane, J. (1964). **Learning curve approach to reliability monitoring**. *IEEE Transactions on Aerospace*, 2(2) :563–566.

[Feng et al., 2011] Feng, Y., Tang, S., et Dai, G. (2011). **Fault tolerant data aggregation scheduling with local information in wireless sensor networks**. *Science and Technology*, 16(5) :451–463.

[Fumera et al., 2005] Fumera, G., et Roli, F. (2005). **A theoretical and experimental analysis of linear combiners for multiple classifier systems**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6) :942–956.

[Gabriel et al., 1969] Gabriel, K., et Sokal, R. (1969). **A new statistical approach to geographic variation analysis**. *Systematic Zoology*, 18 :259âĂŞ278.

[Gallais et al., 2006] Gallais, A., Carle, J., Simplot-Ryl, D., et Stojmenovic, I. (2006). **Localized sensor area coverage with low communication overhead**. Dans *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 328–337.

[Geeta et al., 2013] Geeta, D., Nalini, N., et Biradar, R. C. (2013). **Fault tolerance in wireless sensor network using hand-off and dynamic power adjustment approach**. *Journal of Network and Computer Applications*, 36(4) :1174–1185.

[Goode et al., 2000] Goode, K., Moore, J., et Roylance, B. (2000). **Plant machinery working life prediction method utilizing reliability and condition-monitoring data**. *Proceedings of the IMechE, PartE : Journal of Process Mechanical Engineering*, 214(E2) :109–122.

[Gungor et al., 2006] Gungor, V. C., et Ozgur B. Akan, O. B. (2006). **Dst : Delay sensitive transport in wireless sensor networks**. Dans *Proceedings of the 7th IEEE International Symposium on Computer Networks (ISCN'06)*, pages 116–122.

[Haug, 2005] Haug, A. (2005). **A tutorial on bayesian estimation and tracking techniques applicable to nonlinear and non-gaussian process**. Dans *The Mitre Corporation*, pages 1–52, McLean, Virginia.

[He et al., 2012a] He, S., Chen, J., Cheng, P., Gu, Y., He, T., et Sun, Y. (2012a). **Maintaining quality of sensing with actors in wireless sensor networks**. *IEEE Transactions on Parallel and Distributed Systems*, 23(9) :1657–1667.

[He et al., 2012b] He, S., Chen, J., Li, X., Shen, X. S., et Sun, Y. (2012b). **Leveraging prediction to improve the coverage of wireless sensor networks**. *IEEE Trans. Parallel Distrib. Syst.*, 23(4) :701–712.

[He et al., 2012c] He, S., Chen, J., Li, X., Shen, X. S., et Sun, Y. (2012c). **Leveraging prediction to improve the coverage of wireless sensor networks**. *IEEE Trans. Parallel Distrib. Syst.*, 23(4) :701–712.

[He et al., 2012d] He, S., Chen, J., Yau, D. K. Y., Shao, H., et Sun, Y. (2012d). **Energy-efficient capture of stochastic events under periodic network coverage and coordinated sleep**. *IEEE Trans. Parallel Distrib. Syst.*, 23(6) :1090–1102.

[Hefeeda et al., 2010] Hefeeda, M., et Ahmadi, H. (2010). **Energy-efficient protocol for deterministic and probabilistic coverage in sensor networks**. *IEEE Trans. Parallel Distrib. Syst.*, 21(5) :579–593.

[Heinzelman et al., 2000] Heinzelman, W. R., Chandrakasan, A., et Balakrishna, H. (2000). **Energy-efficient communication protocol for wireless sensor networks**. Dans *IEEE Proceedings of the Hawaii International Conference on System Sciences*.

[Heng et al., 2009] Heng, A., Zhang, S., Tan, A. C., et Mathew, J. (2009). **Rotating machinery prognostics : State of the art, challenges and opportunities**. *Mechanical Systems and Signal Processing*, 23 :724–739.

[Herzog et al., 2009] Herzog, M., Marwala, T., et Heyns, P. (2009). **Machine and component residual life estimation through the application of neural netwoks**. *Reliability Engieneering and System Safety*, 94(2) :479–489.

[Ho, 1998] Ho, T. K. (1998). **The random subspace method for constructing decision forests**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8) :832–844.

[Hu et al., 2012] Hu, C., Youn, B. D., Wang, P., et Yoon, J. T. (2012). **Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life**. *Reliability Engieneering and System Safety*, 103 :120–135.

[Huang et al., 2007] Huang, R., Xi, L., Li, X., Liu, C. R., Qiu, H., et Lee, J. (2007). **Residual life prediction for ball bearings based on self-organizing map and back propagation neural network methods**. *Mechanical Systems and Signal Processing*, 21(1) :193–207.

[ISO13381-1, 2004] ISO13381-1 (2004). **Condition monitoring and diagnostics of machines- prognostics- part1 : General guidelines**.

[Issariyakul et al., 2008] Issariyakul, T., et Hossain, E. (2008). **Introduction to Network Simulator NS2**. Springer Publishing Company, Incorporated, 1 édition.

[Iyer et al., 2005] Iyer, Y. G., Gandham, S., et Venkatesan, S. (2005). **Stcp : A generic transport layer protocol for wireless sensor networks**. Dans *Proceedings of the 14th International Conference on Computer Communications and Networks (ICCCN)*, pages 449–454, San Diego, California, USA.

[Jacques et al., 2007] Jacques, P., Seshagiri, R., Prabhakar, T., Jean-Pierre, H., et Jamadagni, H. (2007). **Commonsense net : a wireless sensor network for resource-poor agriculture in the semiarid areas of developing countries**. *International Journal of Information Tecchnology*, 4(1) :51–67.

[Jardine et al., 2006] Jardine, A. K., Lin, D., et Banjevic, D. (2006). **A review on machinery diagnostics and prognostics implementing condition-based maintenance**. *Mechanical Systems and Signal Processing*, 20 :1483–1510.

[Jina et al., 2008] Jina, Y., Wanga, L., Kimb, Y., et Eemc, X. Y. (2008). **An energyefficient multi-level clustering algorithm for large-scale wireless sensor networks**. *Computer Networks*, 52 :542âĂŞ562.

[J.M.Noortwijk, 2009] J.M.Noortwijk (2009). **A survey of the application of gamma process in maintenance**. *Reliability Engieneering and System Safety*, 94(1) :2–21.

[Kabashi et al., 2008] Kabashi, A., et Elmirghani, J. (2008). **A technical framework for designing wireless sensor networks for agricultural monitoring in developing countries**. Dans *the International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*, page 395âĂŞ401.

[Kait et al., 2007] Kait, L., Kai, C., Khoshdelniat, R., Lim, S., et Tat, E. (2007). **Paddy growth monitoring with wireless sensor networks**. Dans *International Conference on Intelligent and Advanced Systems (ICIAS)*, page 966âĂŞ970, Kuala Lumpur, Malaysia.

[Kallen et al., 2005] Kallen, M., et van Noortwijk, J. (2005). **Optimal maintenance decisions under imperfect inspection**. *Reliability Engieneering and System Safety*, 90 :177–185.

[Kanemoto et al., 2013] Kanemoto, S., Yokotsuka, N., Yusa, N., et Kawabata, M. (2013). **Diversity and integration of rotating machine health monitoring methods**. Dans *Chemical Engineering Transactions*, numéro 33, pages 169–174, Milan, Italy.

[Kasbekar et al., 2011a] Kasbekar, G. S., Bejerano, Y., et Sarkar, S. (2011a). **Lifetime and coverage guarantees through distributed coordinate-free sensor activation**. *IEEE/ACM Trans. Netw.*, 19(2) :470–483.

[Kasbekar et al., 2011b] Kasbekar, G. S., Bejerano, Y., et Sarkar, S. (2011b). **Lifetime and coverage guarantees through distributed coordinate-free sensor activation**. *IEEE/ACM Trans. Netw.*, 19(2) :470–483.

[Kashi et al., 2012] Kashi, S. S., et Sharifi, M. (2012). **Coverage rate calculation in wireless sensor networks**. *Computing*, 94(14) :833–856.

[Kazmierczak, 1983] Kazmierczak, K. (1983). **Application of autoregressive prognostic techniques in diagnostics**. Dans *The Vehicle Diagnostics Conference*, Tuczno, Poland.

[Ke et al., 2009] Ke, W., Liqiang, W., Shiyu, C., et Song, Q. (2009). **An energy-saving algorithm of wsn based on gabriel graph**. Dans *5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, page 1âĂŞ4, Beijing, China.

[Kim et al., 2006] Kim, M., Doh, I., et Chae, K. (2006). **Denial of service (dos) detection through practical entropy estimation on hierarchical sensor networks**. Dans *The 8th International Conference Advanced Communication Technology ICACT*, volume 3, pages 1566–1571.

[K.Ito et al., 2000] K.Ito, et Xiong, K. (2000). **Gaussian filters for nonlinear filtering problems**. *IEEE Transactions on Automatic Control*, 45(5) :910–927.

[Knight, 2004] Knight, J. C. (2004). **An introduction to computing system dependability**. Dans *Proceedings of the 26th International Conference on Software Engineering (ICSE'04)*.

[Koushanfar et al., 2004] Koushanfar, F., Potkonjak, M., et Sangiovanni-Vincentelli, A. (2004). **Handbook of Sensor Networks : Compact Wireless and Wired Sensing Systems**, chapitre 8, pages 812–836. CRC Press.

[Kwon et al., 2011] Kwon, S., Ko, J. H., Kim, J., et Kim, C. (2011). **Dinamic timeout for data aggregation in wireless sensor netwoks**. *Computer Networks*, 55 :650–664.

[Lee, 1980] Lee, L. (1980). **Testing adequacy of the weibull and log linear rate models for a poisson process**. *Technometrics*, 22(2) :195–199.

[Lee et al., 2008] Lee, M.-H., et Choi, Y.-H. (2008). **Fault detection of wireless sensor networks**. *Computer communications*, 31 :3469–3475.

[Li et al., 2011] Li, Z., et Gong, G. (2011). **A survey on security in wireless sensor netwoks**. Rapport technique, Department of Electrical and Computer Engineering, University of Waterloo, Canada.

[Marquez et al., 2005] Marquez, A. C., Heguedas, A. S., et Iung, B. (2005). **Monte carlo-based assessment of system availability. a case study for cogeneration plants**. *Reliability Engieneering and System*, 88 :273–289.

[Matula et al., 1980] Matula, D., et Sokal, R. (1980). **Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane**. *Geographical Analysis*, 12(3) :205âĂŞ222.

[Medjaher et al., 2012] Medjaher, K., Tobon-Mejia, D., et Zerhouni, N. (2012). **Remaining useful life estimation of critical components with application to bearings**. *IEEE Transactions on Reliability*, 61(2) :292–302.

[Mojoodi et al., 2011] Mojoodi, A., Mehrani, M., Forootan, F., et Farshidi, R. (2011). **Redundancy effect on fault tolerance in wireless sensor networks**. *Global Journal of Computer Science & Technology*, 11(6) :34–39.

[Niu et al., 2010] Niu, G., et Yang, B.-S. (2010). **Intelligent condition monitoring and prognostics system based on data-fusion strategy**. *Expert Systems with Applications*, 37 :8831–8840.

[Parno et al., 2005] Parno, B., Perrig, A., et Gligor, V. (2005). **Distributed detection of node replication attacks in sensor networks**. Dans *IEEE Sympsium on Security and Privacy*.

[Peng et al., 2010] Peng, Y., Dong, M., et Zuo, M. J. (2010). **Current status of machine prognostics in condition-based maintenance : a review**. *International journal of Advanced Manifacturing Technology*, 50 :297–313.

[Russell et al., 2003] Russell, S. J., et Norvig, P. (2003). **Artificial Intelligence : A Modern Approach**. Pearson Education, 2 édition.

[Schapire, 1999] Schapire, R. E. (1999). **A brief introduction to boosting**. Dans *Proceedings of the sixteenth International Joint Conference on Artificial Intelligence*.

[Sharkey et al., 1997] Sharkey, A., et Sharkey, N. (1997). **Combining diverse neural nets**. *The Knowledge Egineering Review*, 12(3) :231–247.

[Sikorska et al., 2011] Sikorska, J., Hodkiewicz, M., et Ma, L. (2011). **Prognostic modelling options for remaining useful life estimation by industry**. *Mechanical Systems and Signal Processing*, 25 :1803–1836.

[Silva et al., 2012] Silva, I., Guedes, L. A., Portugal, P., et Vasques, F. (2012). **Reliability and availability evaluation of wireless sensor netwoks for industrial applications**. *Sensors*, 12 :806–838.

[Solis et al., 2004] Solis, I., et Obraczka, K. (2004). **The impact of timing in data aggregation for wireless sensor networks**. Dans *Proceedings of the IEEE International Conference on Communications*, page 3640âĂŞ3645.

[Taherkordi et al., 2006] Taherkordi, A., Taleghan, M. A., et Sharifi, M. (2006). **Dependability considerations in wireless sensor netwoks applications**. *Journal of Networks*, 1(6) :28–35.

[Thurston, 2001] Thurston, M. (2001). **An open standard for web-based condition-based maintenance systems**. Dans Conference, I. S. R. T., éditeur, *AUTOTESTCON Proceedings*, pages 401–415.

[Tian et al., 2005] Tian, D., et Georganas, N. D. (2005). **Connectivity maintenance and coverage preservation in wireless sensor networks**. *Ad Hoc Networks*, 3 :744–761.

[Tobon-Mejia et al., 2012a] Tobon-Mejia, D., Medjaher, K., et Zerhouni, N. (2012a). **Cnc machine tool's wear diagnostic and prognostic by using dynamic bayesian networks**. *Mechanical Systems and Signal Processing*, 28 :167–182.

[Tobon-Mejia et al., 2012b] Tobon-Mejia, D. A., Medjaher, K., Zerhouni, N., et Tripot, G. (2012b). **A data-driven failure prognostics method based on mixture of gaussians hidden markov models**. *IEEE Transactions on Reliability*, 61(2) :491–503.

[Todinov, 2005] Todinov, M. (2005). **Reliability and risk models–setting reliability requirements**. Rapport technique, John Wiley & Sons Ltd, Chichester England.

[Toscano et al.,] Toscano, E., Mirabella, O., et Bello, L. L. **An energy-efficient realtime communication framework for wireless sensor networks**.

[Tsui et al., 1995] Tsui, F., Sun, M., Li, C., et Sclabassi, A. (1995). **Wavelet-based neural netwok for prediction of icp signal**. Dans *IEEE Engineering in Medecine and Biology*, pages 1045–1046, Montreal, Canada.

[Tsymbal et al., 2000] Tsymbal, A., et Puuronen, S. (2000). **Bagging and boosting with dynamic integration of classifiers**. Dans *The 4th European Conference on Principles and Practice of Knowledge Discovery in Data Bases PKDD*, pages 116–125.

[Tumer et al., 1996] Tumer, K., et Ghosh, J. (1996). **Error correlation and error reduction in ensemble classifiers**. *Connection Science*, 8 :385–404.

[Verikas et al., 2011] Verikas, A., Gelzinis, A., et Bacauskiene, M. (2011). **Mining data with random forests : A survey and results of new tests**. *Pattern Recognition*, 44.

[Walters et al., 2007] Walters, J. P., Liang, Z., Shi, W., et Chaudhary, V. (2007). **Wireless sensor network security : A survey**. Dans *Security in Distributed, Grid and Pervasive Computing*, pages 799–849. CRC Press.

[Wang et al., 2004] Wang, W., Golnaraghi, M., et Ismail, F. (2004). **Prognostics of machine health condition using neuro-fuzzy systems**. *Mechanical Systems and Signal Processing*, 18 :813–831.

[Wang et al., 2003] Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., et Gill, C. (2003). **Integrated coverage and connectivity configuration in wireless sensor netwoks**. Dans *First ACM Conference on Embedded Networked Systems*.

[Weidl et al., 2003] Weidl, G., Madsen, A., et Dahlquist, E. (2003). **Object-oriented bayesian netwoks for industrial process operation**. Dans *Bayesian Modelling Applications Workshop Associated with the 19th Conference on Uncertainties in Artificial Intelligence*, pages 1–9, Acapulco, Mexico.

[Weidl et al., 2005] Weidl, G., Madsen, A., et Israelson, S. (2005). **Applications of object-oriented bayesian networks for condition monitoring, root cause analysis and desicion support on operation of complex continuous process**. *Computers and Chemical Engineering*, 29 :1996–2009.

[Wood et al., 2002] Wood, A. D., et Stankovic, J. A. (2002). **Denial of service in sensor networks**. *Computer*, 35(10) :54–62.

[Wu et al., 2007] Wu, W., Hu, J., et Zhang, J. (2007). **Prognostics of machine healt condition using an improved arma-based prediction method**. Dans *IEEE*, pages 1062–1067, China.

[Yan et al., 2004] Yan, J., Koc, M., et Lee, J. (2004). **A prognostic algorithm for machine performance assessment and its application**. *Production Planning and Control*, 76 :796–801.

[Yang et al., 2007] Yang, W., Liusheng, H., Junmin, W., et Hongli, X. (2007). **Wireless sensor networks for intensive irrigated agriculture**. Dans *the Consumer Communications and Networking Conference (CCNC)*, page 197âĂŞ201, Las Vegas, NV, USA.

[Ye et al., 2003] Ye, F., Zhong, G., Cheng, J., Lu, S., et Zhang, L. (2003). **Peas : A robust energy conserving protocol for long-lived sensor networks**. Dans *Proceedings of the 23rd International Conference on Distributed Computing Systems*, ICDCS'03, pages 28–37.

[YE et al., 2008] YE, W., HEIDEMANN, J., et ESTRIN, D. (2008). **An energy-efficient mac protocol for wireless sensor networks**. *Wireless Sensor Network*, 1 :1–69.

[Yoo et al., 2007] Yoo, S., Kim, J., Kim, T., Ahn, S., Sung, J., et Kim, D. (2007). **A2s : automated agriculture system based on wsn**. Dans *the IEEE International Symposium on Consumer Electronics (ISCE)*, page 1âĂŞ5, Dallas, TX, USA,.

[Yousefi et al., 2012] Yousefi, H., Yeganeh, M. H., Alinaghipour, N., et Movaghar, A. (2012). **Structure-free real-time data aggregation in wireless sensor networks**. *Computer Communications*, 35(9) :1132âĂŞ1140.

[Zadeh, 1965] Zadeh, L. A. (1965). **Fuzzy sets**. *Information and control*, 8 :338–353.

[Zhang et al., 2012] Zhang, L., Ye, Q., Cheng, J., Jiang, H., Wang, Y., Zhou, R., et Zhao, P. (2012). **Fault-tolerant scheduling for data collection in wireless sensor networks**. Dans *Proceedings of IEEE GLOBECOM*.

[Zhang et al., 2014] Zhang, Q., Fu, L., Gu, Y., Gu, L., Cao, Q., Chen, J., et He, T. (2014). **Collaborative scheduling in highly dynamic environments using error inference**. *IEEE Transactions on Parallel and Distributed Systems*, 25(3) :591–601.

[Zhang et al., 2005] Zhang, Q., Wang, P., Reeves, D. S., et Ning, P. (2005). **Defending against sybil attacks in sensor networks**. Dans *25th IEEE International Conference on Distributed Computing Systems Workshops*, pages 185–191.

[Zhou et al., 2005] Zhou, Y., Lyu, M. R., Liu, J., et Wang, H. (2005). **Port : A price-oriented reliable transport protocol for wireless sensor networks**. Dans *Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*.

[Zio et al., 2010] Zio, E., et Maio, F. D. (2010). **A data-driven fuzzy appoach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system**. *Reliability Engieneering and System Safety*, 95 :49–57.

[Zorbas et al., 2007] Zorbas, D., Glynos, D., et Douligeris, C. (2007). **BGOP : An adaptive algorithm for coverage problems in wireless sensor netwoks**. Dans *the 13th European Wireless Conference*.

Document réalisé avec LaTeX et :
le style LaTeX pour Thèse de Doctorat créé par S. Galland — http://www.multiagent.fr/ThesisStyle
la collection de paquets `tex-upmethodology` — http://www.arakhne.org/tex-upmethodology/

Résumé :

Une maintenance prédictive efficace se base essentiellement sur la fiabilité des données de surveillance. Dans certains cas, la surveillance des systèmes industriels ne peut pas être assurée à l'aide de capteurs individuels ou filaires. Les Réseaux de Capteurs Sans Fil (RCSF) sont alors une alternative. Vu la nature de communication dans ces réseaux, la perte de données est très probable. Nous proposons un algorithme distribué pour la survie des données dans le réseau. Cet algorithme réduit le risque d'une perte totale des paquets de données et assure la continuité du fonctionnement du réseau. Nous avons aussi simulé de différentes topologies du réseau pour évaluer leur impact sur la complétude des données au niveau du nœud puits. Par la suite, nous avons proposé une démarche d'évaluation de l'état de santé de systèmes physiques basée sur l'algorithme des forêts aléatoires. Cette démarche repose sur deux phases : une phase hors ligne et une phase en ligne. Dans la phase hors ligne, l'algorithme des forêts aléatoires sélectionne les paramètres qui contiennent le plus d'information sur l'état du système. Ces paramètres sont utilisés pour construire les arbres décisionnels qui constituent la forêt. Dans la phase en ligne, l'algorithme évalue l'état actuel du système en utilisant les données capteurs pour parcourir les arbres construits. Chaque arbre dans la forêt fournit une décision, et la classe finale est le résultat d'un vote majoritaire sur l'ensemble de la forêt. Quand les capteurs commencent à tomber en panne, les données décrivant un indicateur de santé deviennent incomplètes ou perdues. En injectant de l'aléatoire dans la base d'apprentissage, l'algorithme aura des points de départ différents, et par la suite les arbres aussi. Ainsi, l'absence des mesures d'un indicateur de santé ne conduit pas nécessairement à l'interruption du processus de prédiction de l'état de santé.

Mots-clés : Réseaux de capteurs sans fil, Evaluation de l'état de santé, Tolérance aux pannes, Consommation d'énergie, Forêts aléatoires, Topologies réseau.

Abstract:

An efficient predictive maintenance is based on the reliability of the monitoring data. In some cases, the monitoring activity cannot be ensured with individual or wired sensors. Wireless sensor networks (WSN) are then an alternative. Considering the wireless communication, data loss becomes highly probable. Therefore, we study certain aspects of WSN reliability. We propose a distributed algorithm for network resiliency and data survival while optimizing energy consumption. This fault tolerant algorithm reduces the risks of data loss and ensures the continuity of data transfer. We also simulated different network topologies in order to evaluate their impact on data completeness at the sink level. Thereafter, we propose an approach to evaluate the system's state of health using the random forests algorithm. In an offline phase, the random forest algorithm selects the parameters holding more information about the system's health state. These parameters are used to construct the decision trees that make the forest. By injecting the random aspect in the training set, the algorithm (the trees) will have different starting points. In an online phase, the algorithm evaluates the current health state using the sensor data. Each tree will provide a decision, and the final class is the result of the majority vote of all trees. When sensors start to break down, the data describing a health indicator becomes incomplete or unavailable. Considering that the trees have different starting points, the absence of some data will not necessarily result in the interruption of the prediction process.

Keywords: Wireless sensor networks, Health assessment, Fault tolerance, Energy consumption, Random forests, Network topologies.

SPIM

■ École doctorale SPIM 16 route de Gray F - 25030 Besançon cedex
■ tél. +33 (0)3 81 66 66 02 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

UFC UNIVERSITÉ DE FRANCHE-COMTÉ