Information Fusion in Distributed Sensor Networks with Byzantines

Andrea Abrardo, Mauro Barni, Kassem Kallas, Benedetta Tondi

Springer Signals and Communication Technology, 2019

Siena
December, 2019

Contents

List of Figures

List of Tables

List of Symbols

H_0	null hypothesis
H_1	alternative hypothesis
n	number of nodes in the network
$\mathbf{x}_i$	observation vectors available to sensor i
S_i	the system state under hypothesis $H_i, i \in \{0, 1\}$
$P(H_0)$	a-prior probability that the system is in state S_0
$P(H_1)$	a-prior probability that the system is in state S_1
$P(x\|H_j)$	the observation probability density conditioned to hypothesis H_j
$S^* \in \{0, 1\}$	the global decision at the fusion center regarding S
C_{ij}	cost of deciding H_i when H_j is true
$\mathcal{C}$	average cost or risk function for Bayesian detection
$\Lambda(x)$	likelihood ratio regarding the observation x
λ	decision threshold

P_{FA}	probability of false alarm
P_{MD}	probability of missed detection
P_D	probability of correct detection
P_{null}	probability to decide H_0 when H_0 is true
P_e	probability of error
λ_{NP}	local Neyman-Pearson likelihood decision threshold
α_{NP}	acceptable false alarm for Neyman-Pearson detector
$\mathcal{F}$	Lagrange function for Neyman-Pearson detector optimization
$\lambda_i, i \in \{0, 1\}$	decision threshold for hypothesis H_i for local SPRT detector
α_{ST}	local SPRT detector constraint on false alarm probability
β_{ST}	local SPRT detector constraint on missed detection probability
u_i	information sent by sensor i to the FC
P_{d_i}	local probability of correct detection at node i
P_{fa_i}	local probability of false alarm at node i
P_{md_i}	local probability of missed detection at node i
Q_D	global probability of correct detection at the FC
Q_{FA}	global probability of false alarm at the FC
$Q_{D_{AND}}$	global probability of correct detection for the AND rule
$Q_{FA_{AND}}$	global probability of false alarm for the AND rule
$Q_{D_{OR}}$	global probability of correct detection for the OR rule
$Q_{FA_{OR}}$	global probability of false alarm for the OR rule

$Q_{D_{kn}}$	global probability of correct detection for the k-out-of-n rule
$Q_{FA_{kn}}$	global probability of false alarm for the k-out-of-n rule
U_{SLC}	square Law Combining information fusion result
U_{MRC}	maximum Ratio Combining information fusion result
U_{SC}	selection Combining information fusion result
ζ	decision threshold of the soft combination rules
$\Upsilon_i, i \in \{0,1\}$	decision threshold for hypothesis H_i for global SPRT detector
α_{FC}	global SPRT detector constraint on false alarm probability
β_{FC}	global SPRT detector constraint on missed detection probability
$\mathcal{G} = (\mathcal{N}, \mathcal{E})$	graph $\mathcal{G}$ with the set of vertices $\mathcal{N} = \{n_1, ..., n_n\}$ and the set of edges $\mathcal{E}$
$\mathcal{N}_i$	neighborhood list of node i
$x_i(k)$	state value of node i at iteration k of the consensus algorithm
w_i	weight assigned to node i state update value
ϵ	consensus update parameter
$\bar{x}$	consensus value or the average value of all initial state values
$\mathcal{X}^2_M$	chi-square distribution with M degrees of freedom
$\Gamma(.)$	the incomplete gamma function
$Q(.)$	the generalized Marcum *Q-function*
γ_S	the SNR experienced by the Energy Detector

$\mathcal{S}_i$	strategy set available to player i
v_l	payoff (or utility) of player l
$G(N, \mathcal{S}, \mathbf{v})$	game with N players, strategy set $\mathcal{S}$ and payoff vector $\mathbf{v}$
ss	strategy profile vector for a game with N players
$\Pi(\mathcal{Z})$	set of all probability distributions over the set $\mathcal{Z}$
r_i	report sent by node i to the FC
α	fraction of nodes (or links) under attack or the probability that a node (or link) is under attack
$\tilde{\mathbf{x}}_i$	attacked observation seen by node i
$\tilde{x}_i(0)$	falsified initial data of node i in consensus network
$\tilde{w}_i$	tampered weight for node i's state value in consensus network
Δ_i	attack value in falsification attack on consensus network
$u_i(k)$	attack value of consensus disruption attack at node i for the $k+1$ iteration step
w_i	weight assigned to node i state update value
r_{ij}	report by node i at instant j
m	observation window size
P_{mal}	node malicious probability or crossover probability of the attacked links
u_{ij}	decision by node i at instant j
Γ_i	hard reputation score of node i
$d_{int}(j)$	intermediate decision at instant j at the FC

η	isolation threshold
R_{ij}	soft reputation score of node i at instant j
$DF(\mathcal{S}_{FC}, \mathcal{S}_{FC}, v)$	decision fusion game with $\mathcal{S}_{FC}$ the strategy set for the FC, $\mathcal{S}_B$ the strategy set for Byzantines, and payoff v
$P_{e,ar}$	probability of error after removal of Byzantines
P_{ISO}^{B}	probability of correct isolation of Byzantines
P_{ISO}^{H}	probability of erroneous isolation of honest nodes
P_{mal}^{FC}	the FC guess of P_{mal}
$P_X(x)$	probability mass function of the random variable x
S^m	sequence of system states random variable with instantiation s^m
$P_{S_j}(i), i \in \{0, 1\}$	probability that a system is in state S_j at time i
U_{ij}	random variable for the local decision of node i at instant j with instantiation u_{ij}
$A^n = (A_1, \ldots, A_n)$	binary random sequence for Byzantines positions with a^n its instantiation
$\mathbf{R} = \{R_{ij}\}$	random matrix of all received reports by FC with $\mathbf{r} = \{r_{ij}\}$ as its instantiation
$P(a^n)$	probability of Byzantines sequence
ε	local decision error at the nodes
δ	the probability that the FC receives a wrong report
$m_{eq}^{(i)}$	the number of instants at which the report is equal to the system state for node i

$E[N_B]$	expected number of Byzantines
μ_{A_i}	expected value of A_i
$H(A^n)$	entropy distribution of Byzantines
$h(\mu_{A_i})$	binary entropy function for the expected value of A_i
h	the FC expected maximum number of Byzantines
$\mathcal{I} = \{1, \ldots, n\}$	indexing set of size n
$\mathcal{I}_k$	set of all k-subsets of $\mathcal{I}$
I	random variable with indexes of byzantine nodes
$P(I)$	equivalent to the probability of a Byzantine sequence$P(a^n)$
n_B	fixed number of Byzantines in the network known to the FC
$DF_{Byz}(\mathcal{S}_B, \mathcal{S}_{FC}, v)$	decision fusion game with $\mathcal{S}_B$ the strategy set of Byzantines, $\mathcal{S}_{FC}$ the strategy set of the FC, and v the payoff
P^B_{mal}	malicious probability strategy of the Byzantines
$\mathcal{S}^q_B$	quantized Byzantines's strategy set
$\mathcal{S}^q_{FC}$	quantized FC's strategy set with $\mathbf{r} = \{r_{ij}\}$ as its instantiation
$\mathbf{V}$	payoff matrix for each pair of strategies
P^*_e	probability of error at the equilibrium
$P(P^B_{mal})$	probability assigned by Byzantines to a strategy in mixed strategy Nash equilibrium
$P(P^{FC}_{mal})$	probability assigned by FC to a strategy in mixed strategy Nash equilibrium
ρ	state transition probability in a two-state markov model

$m_{vf}^{(l)}$	variable-to-function message for factor l
$m_{fv}^{(l)}$	function-to-variable message for factor l

Acknowledgements

WE would like to

Siena
31/12/2019

Abstract

Every *day we share our personal information through digital systems which are constantly exposed to threats. For this reason, security-oriented disciplines of signal processing have received increasing attention in the last decades: multimedia forensics, digital watermarking, biometrics, network monitoring, steganography and steganalysis are just a few examples. Even though each of these fields has its own peculiarities, they all have to deal with a common problem: the presence of one or more adversaries aiming at making the system fail. Adversarial Signal Processing lays the basis of a general theory that takes into account the impact that the presence of an adversary has on the design of effective signal processing tools.*

By focusing on the application side of Adversarial Signal Processing, namely adversarial information fusion in distributed sensor networks, and adopting a game-theoretic approach, this book presents the recent advances in the field and how several issues has been addressed. First, a heuristic decision fusion setup is presented together with the correspondent soft isolation defense scheme that protects the network from adversaries, specifically, Byzantines. Second, the development of an optimum decision fusion strategy in the presence of Byzantines is delineated. In the next step, a technique to reduce the complexity of the optimum fusion by relying on a novel nearly-optimum message passing algorithm based on factor graphs is presented. Afterwards, the message passing approach is considered under Hidden-Markov observations and in the presence of synchronized attacks. Finally, we present the message

passing approach with unbalanced a priors and the synchronized asymmetric attacks.

Chapter 1

Introduction

1.1 Motivation

In the era of digital revolution, intelligent and digital systems are invading our lives. This evolution has a fundamental impact on social, political and economical domains both at personal and society level.

While this digital world is of extreme importance and contributes to the health of our society, its ultra-fast growth creates new opportunities to perpetrate digital crimes, that is cybercrimes, all the more, that this new kind of criminal activity does not need anymore the physical presence of criminals on the crime scene. Criminals and victims are no more limited to territorial borders since crimes are perpetrated in a virtual cyberspace. These crimes can target economy and finance, public health and national security.

Cybercrimes encompass a spectrum of activities ranging from violating personal privacy to illegal retrieval of digital information about a firm, a person and so on. Crimes like fraud, child pornography, violation of digital privacy, money laundering, and counterfeiting stand on the middle of the cybercrimes spectrum. Due to the anonymity provided by the internet, criminals are concealed over the cyberspace to attack their specific victims. Another part of these crimes aims at altering the data of individuals within corporations or government bureaucracies for either profit or political objectives. The other part of the spectrum is occupied by crimes that aim at disrupting internet functionality. These range from spam, hacking, and Denial of Service (DoS) attacks, to cyberterrorism that, according to the U.S. Federal Bureau of Investigation (FBI), is any *"premeditated, politically motivated attack against information, computer systems, computer programs, and data which results in violence against non-combatant targets by sub-national groups or clandestine*

agents". The public awareness about the danger of cyberterrorism has grown dramatically since 11 September 2001 attacks. Especially nowadays, due to the existence of numerous terrorist groups that are interested in attacking a vast list of targets. These groups are benefiting from the cyberspace to recruit personnel to get involved into terrorist activities. As an evidence about the economical impact of cybercrimes, McAfee reported in 2014 that the estimated cost of cybercrime on the global economy is more than 400 billion dollars [1].

For all these reasons, the fight against cybercrime occupies a top position in the priorities list of many governments around the globe. For instance, in the U.S., the FBI's Cyber Division is the agency responsible to combat cybercrime [2]. Other agencies like the U.S. Secret Service (USSS) and U.S. Immigration and Customs Enforcement (ICE) have specific branches committed to fight cybercrimes [3]. Moreover, the USSS runs the National Computer Forensic Institute (NCFI) that offers training courses in computer forensics. [4].

Attention to and fight against cybercrimes is not limited to governmental institutions, it also involves scientific researchers with various backgrounds. Researchers devote their effort to develop effective solutions to security problems and take effective steps toward secure defense solutions and algorithms to combate cybercrime. Signal processing researchers are on top of the list of scientists engaged in such an effort. Increasing attention has been devoted to disciplines like multimedia forensics [5], [6], digital watermarking [7], steganography and steganalysis [8], biometrics [9], network intrusion detection, spam filtering [10], [11], traffic monitoring [12], videosurveillance [13] and many others. Despite enormous differences, all these fields are characterized by a unifying feature: the presence of one or more adversaries aiming at hindering the success of signal processing operators. So far, the problem of coping with an adversary has been addressed by different communities with very limited interaction among them. It is not surprising, then, that similar solutions are re-invented several times, and that the same problems are faced again and again by ignoring that satisfactory solutions have already been discovered in contiguous fields. While each adversarial scenario has its own peculiarities, there are some common and fundamental problems whose

solution under a unified framework would speed up the understanding of the associated security problems and the development of effective and general solutions.

Adversarial Signal Processing (Adv-SP), a.k.a. Adversary-aware Signal Processing [14] is an emerging discipline that aims at studying signal processing techniques explicitly thought to withstand the intentional attacks of one or more adversaries. Its final aim is modeling the interplay between a Defender, wishing to carry out a certain processing task, and an Attacker, aiming at impeding it. A natural framework to model this interplay relies on Game-Theory which provides a powerful mathematical framework to model the conflict and cooperation between rational decision-makers. This framework helps to overcome the so called "cat & mouse" loop in which researchers and system designers continuously develop new attacks and countermeasures in a never-ending loop. By adopting a game-theoretical formalization, a tremendous step toward the development of the theoretical foundation of Adv-SP has been already made, [5], [15], [16] and [17] are just a few examples. With these ideas in mind, this book presents some general ideas from the Adv-SP field applied to Adversarial Information Fusion in Distributed Sensor Networks. In these networks, some distributed sensors, for instance autonomous sensors, actuators, mobile devices, must provide some information about the state of an observed system. In the centralized approach, the information collected by the sensors is sent to a *"Fusion Center"* (FC). By using all the information received from the nodes, the FC is responsible of making a final global decision about the state of the system of interest. The actual process of integrating the information submitted by several sources into a coherent understanding of the system state is called *"Information Fusion"*. Therefore, Information Fusion, in general, refers to particular mathematical functions, algorithms, methods and procedures for combining information. This term is very flexible and the classification of various techniques depends on several factors like type of information, type of data representation, level of information abstraction, and others [18]. Information fusion techniques are extensively used in several fields. In economics for instance, information fusion is used to compute the Retail Price Index (RPI) which is a measure of the change of the average prices over a certain amount of time, or the Human Development In-

dex (HDI) that is a metric to assess the social and ecomic development levels of countries, and many others. In biology, fusing DNA and RNA sequences is another form of information fusion. Information fusion is used widely in Computer Science and Artificial Intelligence e.g. in robotics for fusion of images in computer vision, ensemble methods for data mining, decision making systems, multi-agents systems and many others [18], [19], [20].

This book focuses on Information Fusion in Distributed Sensor Networks in the presence of adversaries. Specifically, we present a setup wherein some of the sensors might aims at corrupting the information fusion process to pursue an exclusive benefit from the system under inspection, forge the knowledge about the state of the system or corrupt the whole network functionality. These malicious and misbehaving nodes[1] are known as adversaries and due to their presence, the problem at hand is called "Adversarial Information Fusion". Following this setting, the defender or network designer is asked to modify the fusion process to take into account that a part of the network is under the control of the adversaries. The modification of the fusion process is to be implemented at the FC if the network is centralized, while, on the other hand, is to be implemented locally at the nodes when the network is fully decentralized.

Adversarial Information Fusion in Distributed Networks is of great importance in many applications. Cognitive Radio Networks (CRN) offer a first example. The electromagnetic spectrum is a naturally limited resource that is ever-demanded due to the explosion of wireless technology [21], [22]. The use of a fixed access policy is ineffective because it assigns spectrum portions exclusively to licensed users. This raises the need for a more flexibile and efficient spectrum allocation policy. Dynamic Spectrum Access (DSA) is a promising solution to underutilization of the spectrum [23]. In DSA there are two types of users: licensed users known as Primary Users (PU) and second priority users known as Secondary Users (SU). A PU has the highest priority to access the spectrum resource since he is the license holder; SUs on the other hand, are allowed to access the spectrum when it is free or in a shared manner providing no harmful interference to PUs [23]. DSA requires that SUs are

[1] Throughout the book, we will alternatively use the words sensor and node to refer to a distributed sensor network entity.

able to sense, monitor and access the spectrum in an efficient, intelligent and dynamic way. An SU device with cognitive capabilities is known as Cognitive Radio (CR) [24].

Cognitive Radios must sense the spectrum by monitoring the PU activity in order to decide if the spectrum is occupied or not. This decision can be made either locally by exchanging the "measurements" between CRs or remotely by sending the measurements to a FC that "fuses" the received information and broadcasts back the final global decision. The adversaries in CRN can modify their measurements to cause a wrong decision about the spectrum occupancy. This wrong decision can have many effects, for instance, it can cause harmful interference to PU's transmission, exclusive use of the spectrum by the adversaries or even just confusing the network.

Wireless Sensor Networks (WSN) offer another important example. A WSN is a group of spatially distributed sensors that are responsible to monitor a physical phenomenon, e.g. health and environmental conditions like temperature, sound, pressure, etc... In WSN, the sensors are responsibile to measure the physical phenomenon of interest and then pass the gathered information to the nearby nodes or to a FC which fuses all the data received and comes out with a global decision. If the adversary can control some of the sensors, based on its objective and knowledge of the physical system, it can perform arbitrary attacks by flooding the network with random information or devise a strategic attack by sending specific wrong information to force a precise false global decision. These behaviors can disrupt severely the network's functionality and operation and consequently, corrupt the whole WSN.

The emerging field of multimedia forensics offers an additional example. Due to nowdays powerful and user-friendly softwares, editing digital media such as images, video or audio no longer requires professional skills. Typically, editing is used to enhance the media quality, e.g. by enhancing image contrast, denoising an audio track or re-encoding a video to reduce its size. However, altering a digital media can serve less 'innocent' purposes. For instance, to remove or implant evidence or to distribute fake content so to create a deceiving forgery. Multimedia Forensics tackles with this problem based on the observation that any processing tends to leave traces that can

be exploited to expose the occurrence of manipulations [25], [5]. Very often, the creation of a forgery involves the application of more than one single processing tool, thus leaving a number of traces that can be used by the forensic analyst; this consideration suggests to analyze the "authenticity" of digital medias by using more than one tool. Furthermore, these tools are far from ideal and often give uncertain or even wrong answers. Therefore, whenever possible, it is wise to employ more than one tool searching for the same trace. By taking into account the presence of the adversaries, fusing all the local decisions to make a final decision about document's authenticity can improve forgery detection [26].

Online reputation systems are last additional example. A reputation system gathers evidence from agents about objects like products, good, services, business, users or digital contents in order to come out with reputation scores. Most online commercial systems collect user feedbacks as evidence to compute scores for the objects of interest. These scores have a major influence on new online agent's decision. Thus, they provide enough incentive for attackers to manipulate them by providing false/forged feedbacks. By doing so, the attackers try to increase or decrease the reputation of an object and hence, manipulate the decisions of possibile new agents [27].

1.2 Goal and summary

1.2.1 Goal

Various types of adversaries exist for distributed sensor networks and they can be classified depending on many factors: their objectives, their behavior, the amount of information they have about the system under control as well as the network, and so on [28], [29], [30], [31]. Attacks in which adversaries have full control of a number of nodes and behave arbitrarily to disrupt the network are referred to as *Byzantines*. The term Byzantine Attack originates from the Byzantines general problem stated by Lamport et al. in [32] as follows: *"a group of generals of the Byzantine army camped with their troops around an enemy city. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that*

the loyal generals will reach agreement". The relation between this problem and distributed sensor networks is straightforward as Byzantine generals play the role of internal adversaries. A Byzantine adversary may have various behaviors, such as lying about network connectivity, flooding the network with false traffic, forging control information, modifying the information sent by nearby sensors, or controling a strategic set of sensors in the network [33]. As an example, in cooperative spectrum sensing in cognitive radio networks, Byzantine attacks are known as Spectrum Sensing Data Falsification Attack" (SSFD) [34], [35], [36]. In addition to the above examples, in wireless sensor networks, various byzantine attacks exist and they can severly degrade the network performance and corrupt its proper functionality [37], [33].

This book considers the problem of byzantine attacks in distributed sensor network in a binary decision setup [38], [39]. In the literature binary decision is also referred to as binary detection (or Binary Hypothesis Test), since, in many applications, the decision problem refers to the detection of the presence or absence of a certain phenomenon or signal. For instance, in source identification in multimedia forensics, the analyst aims to distinguish which between two sources (e.g. a photo camera and a scanner) generated a specific digital content or to identify the specific device used to acquire the content [5]. In Cognitive Radio Networks, the FC or the network wants to distinguish between two cases: the presence or absence of PU signal in the spectrum of interest [40], [24]. In addition, in WSN, the network wants to detect the presence of a certain physical phenomenon [41]. As a last example, machine learning binary detection and classification for various applications (e.g. spam filtering) to differentiate to which class a specific data belongs [42], [43].

The goal of this book is to recount the present advances in the field of adversarial information fusion in distributed sensor networks, presenting different kinds Byzantine attacks on the binary detection problem and the corresponding defense strategies that mitigate the effect of the attack during the information fusion process. Later, by assuming that Byzantines are not naïve and because *"to every action, there is always a reaction"* [44], a Game-Theoretic approach that follows the Adv-SP setup between the Byzantines (the Attackers) and the Defender (the network designer) is presented.

1.2.2 Summary of the book

In Chapter 4, the book starts by discussing a heuristic adversarial decision fusion setup in which the nodes send to the FC a vector of binary decisions about the state of a system over an observation window. Considering this setup, a soft identification and isolation scheme to exclude the reports sent by the Byzantines from the decision fusion process is discussed. The *isolation* process is the process whereby the FC removes Byzantines's decisions from the fusion process after identifying them among the nodes in the distributed network. By adopting this soft scheme, the FC can assign a reliability value to each node. Moreover, the competition between the Byzantines and the FC is formalized in a game-theoretic sense in order to study the existence of an equilibrium point for the game. Then, the payoff in terms of the decision error probability when the players "play" at the equilibrium is computed.

The optimum decision fusion rule in the presence of Byzantines in a centralized setup is discussed in a well-defined scenario in Chapter 5. By observing the system over an observation window, we derive the Maximum A Posteriori Probability (MAP) rule, while assuming that the FC knows the attack strategy of the Byzantines and their distribution across the network. With regard to the knowledge that the FC has about the distribution of the Byzantines over the network, several cases are considered. First, we present an unconstrained maximum entropy scenario in which the uncertainty about the distribution of Byzantines is maximum, which means that the a-priori infromation available at the FC about the Byzantines's distribution is minimum. Furthermore, we present a more favorable scenario to the FC in which the maximum entropy case is subject to a constraint. In this scenario, the FC has more a-priori information about Byzantines's distribution i.e the average or the maximum number of Byzantines in the network. Finally, we consider the most favorable situation in which the FC knows the exact number of Byzantines present in the network. Concerning the complexity of the optimal fusion rule, an efficient implementation based on dynamic programming is explained. Thereafter, a game-theoretic framework to cope with the lack of knowledge regarding the Byzantines strategy is introduced. In such a framework, the FC makes a "guess" by selecting arbitrarily a Byzantine's attacking strategy within the optimum fusion rule. By considering the decision error

probability as the payoff, we discuss the performance of the Byzantines as well as the FC at the game equilibrium for several setups when the players adopt their best possible strategies.

In Chapter 6 we consider the complexity of the optimum fusion rule by introducing an efficient message passing approach based on factor graph. A more general model for the observed system in which both independent and Markovian sequences are included is presented. Then, the message passing algorithm is proved to give a near-optimal performance while reducing the complexity from exponential to linear as a function of the observation window size.

In Chapter **??**, we present a Markovian information model for the status with a given transition probability that can be perfectly estimated at the FC. The attacking strategy is revised accordingly. According to the new attacking strategy, the byzantine nodes coordinate their attacks by producing correlated reports, with the aim of mimicking the behavior of the original information and at the same time minimizing the information conveyed to the FC about the sequence of states. In this scenario, a nearly-optimal fusion scheme based on message passing and factor graphs is presented. Experimental results show that, although the detector is able to mitigate the effect of Byzantines, the coordination of the efforts is very harmful and significantly impairs the detection performance.

Finally, we present a scenario in which the two states of the system under observation are not equiprobable in Chapter **??**. In this setup, the Byzantines can not adopt a simple corruption strategy consisting in flipping the local decisions regardless of the estimated state of the system. Doing so, in fact, they would reveal their presence to the fusion center, since their reports would not follow the expected statistics. On its side, the fusion center can exploit the knowledge of the a priori probabilities to improve its decision. In view of the above observations, we first introduce a new corruption strategy for the Byzantines, which permits them to make the statistics of their reports indistinguishable from those of the honest nodes. Later, by adopting from the perspective of the fusion center, we present a nearly-optimum, efficient, fusion strategy based on message passing, to face with the new attack. This is done in the most challenging scenario where the Byzantines are synchronised,

i.e. they share a common source of randomness allowing them to submit wrong reports in a simultaneous way.

Chapter 2

Basic notions of Distributed Detection, Information Fusion and Game Theory

2.1 Introduction

Distributed sensor networks consist of a set of spatially distributed sensors that operate as data collectors or decision makers to monitor a shared phenomenon. This is a common case in many real world situations like air-traffic control, economic and finance, medical diagnosis, electric power networks, wireless sensor networks, cognitive radio networks, online reputation systems, and many others. Usually, in centralized networks, if there are no power, channel, communication or privacy constraints, the sensors can send the full raw information they collect to a FC. However, real life situations are different and several constraints must be considered e.g. when sensors are spatially distributed over a large territorial area, when the channel bandwidth is limited, or even when the sensors are supplied with short life power sources. To address these limitations, sensors must perform some local processing before sending a compressed version of the collected information to the FC. The abstraction level of the information summary can vary a lot. For instance, it can be a quantized set of the raw information, a soft summary statistic like an average or a likelihood value, or even a single information bit.

By means of a fusion rule, the FC integrates the information received from the sensors to make a global decision regarding the system or phenomena under probation. The definition of the fusion rule depends on the a-priori information available about the sensors, the transmission channel and the phenomena as well as the information type provided by the sensors. Fusion rules can be as simple as voting rules or advanced sophisticated statistical rules.

Since the behavior of the sensors in the networks could be different due to noise or intentional acts, the interplay between the sensors or between them and the FC could be modeled using Game Theory, which is a mathematical field that studies the situations of competition and/or cooperation, between decision makers or agents

This chapter is divided into two parts. In the first part, we briefly introduce some basic notions of detection theory and outline some detection techniques used locally at the sensors as well as the corresponding decision strategy. Then, we list some common information fusion techniques that can be employed by the FC in the centralized setup. In the second part, we give an overview of game theory by explaining the basic forms of the games, the solution methodology, and the notion of equilibrium between players.

2.2 Detection Theory

Detection theory is a methodology to model the decision making process in order to decide among various classes of items and how to react to that decision. Making decisions and detecting events is not restricted to human being and other living creatures but it also includes intelligent devices and machines. Detection theory is fundamental in the design of electronic and signal processing systems [38] as it has been applied widely in information systems i.e. in radar systems, digital communications, sensor networks, image processing, bio-medicine, control systems, seismology, cognitive radio networks and many others. The main common objective of detection theory is being able to decide about the existence and the status of a phenomenon or event. For example, a radar system must decide about the presence or absence of an aircraft or any other target, a sensor network has to detect the presence or absence of a natural incident, medical test must detect if a certain disease is present or not, and image processing may aim at detecting the presence of a specific object or feature in an image.

The most common case is when the sensors are faced with a binary detection problem i.e. they must decide about the presence or absence of a phenomenon. In the literature, a widely used synonym for the same problem is binary hypothesis testing. Usually, the two hypotheses are denoted by H_0

and H_1 where, H_0 is called the null hypothesis and represents the absence of the phenomenon of interest, whereas H_1 is called the alternative hypothesis and represents the presence of the phenomenon. A more general situation is when the sensors have to decide between a set of M hypotheses. However, this case is out of the scope of this book. By focusing on the binary detection problem, in the setup considered in this chapter, the system state[1] is observed by a network of n sensors through vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. The sensors can decide about the system state by producing the information $u_1, u_2, \ldots, u_n$ that, depending on the information abstraction, can be an information value or a decision bit. The system state $S_i, i \in \{0, 1\}$ can be in S_0 under hypothesis H_0 and in S_1 under hypothesis H_1. $P(H_0)$ and $P(H_1)$ are the a-priori probabilities that the system is under hypothesis H_0 and H_1, respectively. The sensors are not assumed to communicate with each other and compute their local information independently and send it the FC, which in turn, has to come out with a global decision $S^* \in \{0, 1\}$ regarding the state S_i. The above setup is illustrated in Figure 2.1.

In the following section, we consider the case of a single sensor observing the system through a variable x. The sensor employs a certain detection technique in order to make a decision about the state of the system. We will present the most common techniques to perform binary detection and decision locally at the sensor.

2.2.1 Bayesian Detection

In Bayesian detection, two fundamental pieces of information must be available to the sensors: the a-priori probabilities about the system states $P(H_0)$ and $P(H_1)$, and the observation probability densities conditioned to the hypotheses, namely, $p(x|H_0)$ and $p(x|H_1)$. After the decision in favor of $H_i, i \in \{0, 1\}$, four situations are possible; among them two are correct decisions and the others are erroneous. These cases are shown in Table 2.1.

Each decision is taken by the sensor at a cost C_{ij} referring to the case of deciding H_i while H_j is true. Clearly, the erroneous decision costs C_{01} and C_{10} cost the sensor more than the correct decisions (C_{00} and C_{11}). The cost

[1] from now on, we interchange the use of the phenomenon, the event and the system state to refer to the system under probation.

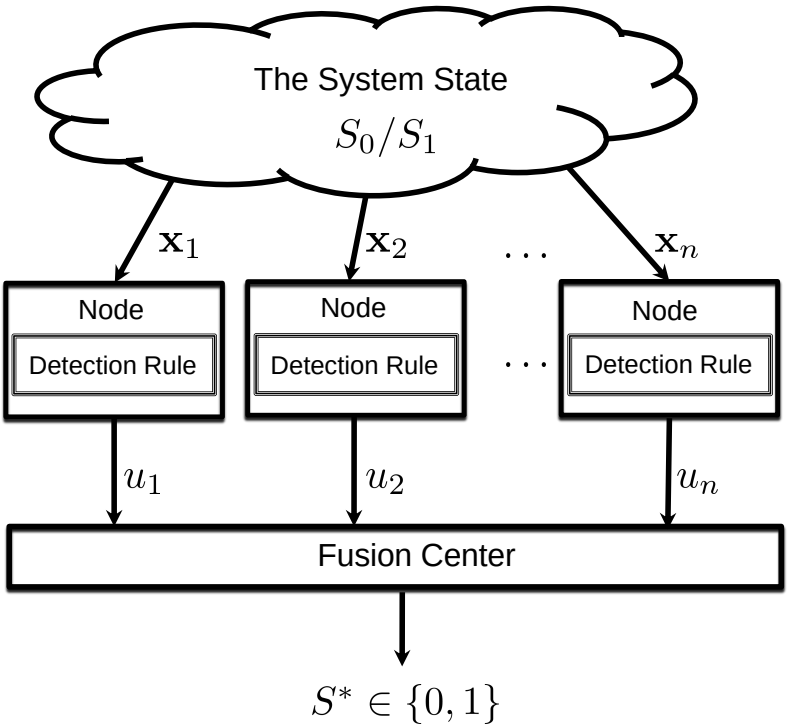


Figure 2.1: *Parallel Topology*

Table 2.1: *Decision cases in binary detection*

	System State H_j	
Decision H_i	$H_0 \vert H_0$	$H_0 \vert H_1$
	$H_1 \vert H_0$	$H_1 \vert H_1$

here refers to the cost of making a decision and the consequences emerging from that decision. Following this formulation, a sensor prefers to employ a decision rule which minimizes the average cost or risk function $\mathcal{C}$ given by

$$\mathcal{C} = \sum_{i=0}^{1} \sum_{j=0}^{1} C_{ij} P(H_i|H_j) P(H_j). \tag{2.1}$$

In [45] it is shown that the decision rule that minimizes $\mathcal{R}$ is given by the

Likelihood Ratio Test (LRT) as

$$\Lambda(x) = \frac{P(x|H_1)}{P(x|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P(H_0)(C_{10} - C_{00})}{P(H_)(C_{01} - C_{11})}, \tag{2.2}$$

where, the left hand side of the equation $\Lambda(x)$ is the likelihood ratio, while the right hand side is the decision threshold λ. By letting,

$$\lambda = \frac{P(H_0)(C_{10} - C_{00})}{P(H_1)(C_{01} - C_{11})}, \tag{2.3}$$

the LRT test can be written as

$$\Lambda(x) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda. \tag{2.4}$$

Consequently, the sensor decides in favor of H_1 when the $\Lambda(x)$ is greater than λ and in favor of H_0 otherwise.

The Log-likelihood ratio test (LLRT) is obtained by applying the logarithm to both sides of Equation (2.4):

$$\log \Lambda(x) \underset{H_0}{\overset{H_1}{\gtrless}} \log \lambda. \tag{2.5}$$

2.2.2 Detection Performance Metrics

For a sensor, the performance of the adopted detection rule is evaluated based on error probabilities of the decision. The two types of error probabilities are the probability of false alarm , P_{fa}, and the probability of missed detection, P_{md}, given by

$$\begin{aligned} P_{fa} &= P(H_1|H_0), \\ P_{md} &= P(H_0|H_1). \end{aligned} \tag{2.6}$$

These terminologies originate from radar theory to indicate the cases of missing an existing target and raising an alarm when the target is absent. A false alarm refers to a case in which the sensor mistakenly decides for H_1 while the true system state is H_0, whereas a missed detection occurs when

the sensor decides for H_0 and the true system state is H_1. In statistics, P_{fa} and P_{md} are known as Type I and Type II error probabilities as well as false positive and false negative, respectively. Consequently, the correct detection probability P_d and the null probability P_{null} (usually called as true negative) are given by

$$\begin{aligned} P_d &= P(H_1|H_1) = 1 - P_{md}, \\ P_{\text{null}} &= P(H_0|H_0) = 1 - P_{fa}. \end{aligned} \tag{2.7}$$

The overall decision error probability is given as

$$P_e = P(H_0)P_{fa} + P(H_1)P_{md}. \tag{2.8}$$

In order to evaluate the performance of a detector, a common graphical representation known as the Receiver Operating Characteristics (ROC) is used [46]. Usually, the ROC curve shows the performance of the detector by plotting the P_d versus P_{fa} by varying the decision threshold. Other forms of the curve are constructed by using other detection probabilities. An example of a ROC

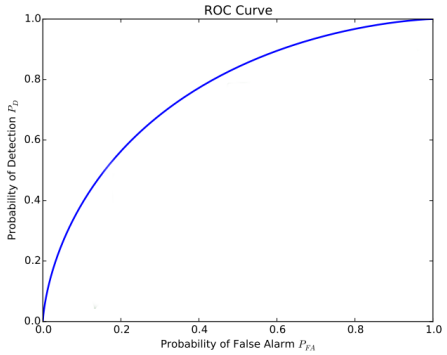


Figure 2.2: *ROC curve example*

curve is depicted in Figure 2.2. From this Figure, it can be seen that the worst case performance of a detector is on the straight line where $P_{fa} = P_d$ since, in this case, the detector is completely "blind" and decides by just flipping a coin. Below this line, the detector can flip its decision to return back to correct decision region. The optimal operation point for a detector is the nearest point to the top left corner of the graph. At this point the detector achieves

the highest P_d with an acceptable constraint on P_{fa} since both probabilities cannot be jointly optimized. An ideal but not realistic detector is a detector with an operating point exactly at that corner with $P_d = 1$ and $P_{fa} = 0$. It is useful to observe that, typically, when a Bayesian detection is used, P_e is considered for performance evaluation while, for the Neyman-Pearson detector presented in the next section, the performance evaluation usually relies on the ROC curve.

2.2.3 Neyman-Pearson Detection

In practice, the a-priori probabilities required to implement a Bayesian detector are difficult to be known. In addition, assigning the costs C_{ij} is difficult or even impossible. Thus, selecting an "optimal" threshold for the LRT test cannot be guaranteed. Neyman-Pearson (NP) detection is a design criterion that overcomes these limitations. By constraining one type of decision error (usually the P_{fa}), the NP detector minimizes the other type. By doing so, the NP detector does not need the a-priori probabilities but instead, it needs to specify a maximum tolerable error for one error type among the two. Formally speaking, the NP detector constraints P_{fa} to an acceptable value α_{NP} and maximizes the detection probability P_d as illustrated in Figure 2.3. Hence, the LRT for the NP setup becomes

$$\log \Lambda(x) \underset{H_1}{\overset{H_0}{\gtrless}} \log \lambda_{NP}. \tag{2.9}$$

The decision threshold λ_{NP} is computed by letting $P_{fa} = \alpha_{NP}$ and solving the following equation:

$$\int_{\lambda_{NP}}^{\infty} p(\Lambda/H_0) d\Lambda = \alpha_{NP}. \tag{2.10}$$

The threshold λ_{NP} obtained by solving this integral, is optimal for the LRT test.

2.2.4 Sequential Detection

In some situations, the sensor decision is based on a vector of observations instead of a single observation and the number of observations needed to

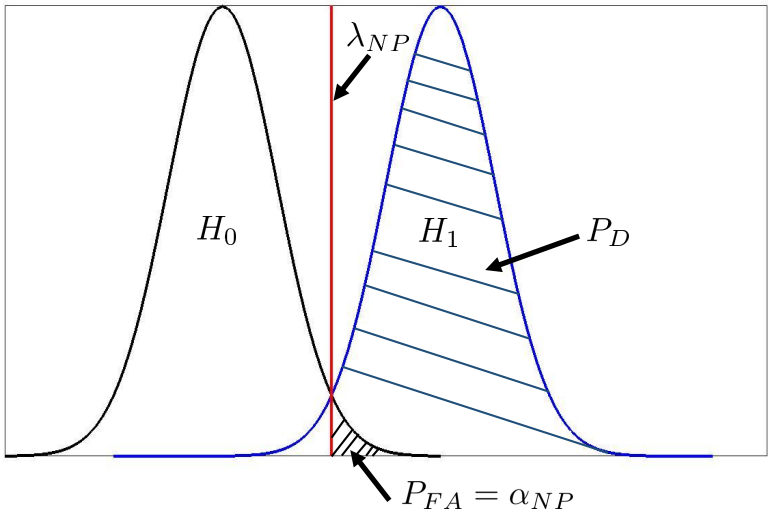


Figure 2.3: *Neyman-Pearson Setup*

make a decision is not fixed. In this situation, the information is gathered sequentially by the sensor over an observation window. To minimize the delay, the sensor make its decision as soon as the collected information is sufficient to make an acceptable "accurate" decision. With sequential detection, a new information is collected only when the available observations are not sufficient to make a decision. For this reason, the sequential detector uses two thresholds for the LRT test, so to collect a new observation only when the LRT value falls between the two thresholds.

The Neyman-Pearson approach to sequential detection has been developed by Abraham Wald and known as the Wald's Sequential Probability Ratio Test (SPRT) [47], [48]. In each step of SPRT, the sensor compares the LRT value to two thresholds λ_0 and λ_1 determined based on pre-defined values for P_{fa} and P_{md}. If the value falls between λ_0 and λ_1, the sensor takes a new observation. On the other hand, the sensor decides for H_1 if the LRT result is greater than λ_1 whereas, it decides for H_0 when the value is smaller than λ_0. The decision scenario of the SPRT detector is depicted in Figure 2.4.

The vector $\mathbf{x}_K = [x_1, x_2, \ldots, x_K]$ represents the observations gathered by a sensor at the K^{th} time instant. Then, the LRT of the SPRT is constructed

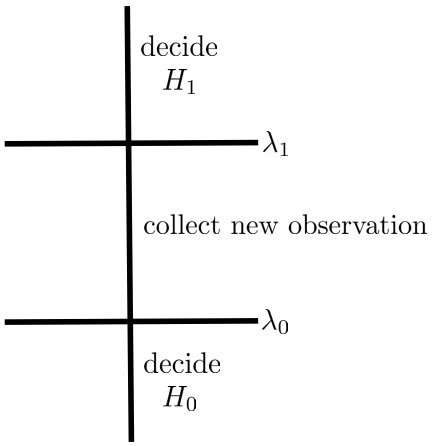


Figure 2.4: *SPRT detector*

as follows

$$\Lambda(\mathbf{x}_K) = \frac{p(\mathbf{x}_K|H_1)}{p(\mathbf{x}_K|H_0)} = \prod_{k=1}^{K} \frac{p(x_k|H_1)}{p(x_k|H_0)}. \tag{2.11}$$

The thresholds λ_0 and λ_1 are computed by constraining P_{fa} and P_{md} to α_{ST} and β_{ST}, respectively, and are computed as follows

$$\begin{aligned} \lambda_0 &\approx \frac{\beta_{ST}}{1-\alpha_{ST}} \\ \lambda_1 &\approx \frac{1-\beta_{ST}}{\alpha_{ST}}. \end{aligned} \tag{2.12}$$

The reason of the approximation is that in the discrete case, the signal may cross the threshold between the samples. Therefore, the approximation is obtained by neglecting the overshoots associated with the threshold crossing events.

The main drawback of this detection method is the time required to make a decision. In addition, the detector can get stuck in a never-ending loop between the thresholds λ_0 and λ_1 due to the low quality of the observations. On the contrary, the advantage of the SPRT is that it takes on average fewer observations than a fixed size observation test [45].

2.3 Information Fusion Rules

In this section, we consider the parallel topology depicted in Figure 2.1. This is, the most common topology to model distributed sensor networks. In the following, we give an overview of the commonly used fusion rules. The choice of a fusion rule depends on many factors; for instance, the processing capability of the sensors, the available information about the system and the network, the channel bandwidth and quality, the energy consumption, the presence of attacks and adversaries and many others.

In centralized networks considered in the rest of the book, after local processing each sensor i sends its information u_i to the FC. Each sensor is assumed to have local detection and false alarm probabilities P_{d_i}, P_{fa_i}. The performance of the fusion rule deployed at the FC is measured using global detection and false alarm probabilities denoted as Q_D and Q_{FA}, respectively. Two classes of fusion rules are considered: simple and advanced rules. In the former, the processing burden is low and the amount of a-priori information required at the FC is small; while in the latter, more processing is required and more information must be known in advance since most of these rules are statistically based.

2.3.1 Simple Fusion Rules

The term "simple" refers to the fact that the operations performed at the FC are computationally cheap. These rules can be used in the absence of a-priori information about the system and the network. We start by considering the case of binary reports, a case which is suitable for bandwidth limited applications. By receiving a pool of binary bits, the FC can apply a "hard" or "voting" fusion rule, namely, AND, OR and k-out-of-n rule [49]. The decision bit is computed locally at the sensor by performing an LRT detection as in Section 2.2, then, the node sends a bit 1 when the LRT decides for H_1 and 0 otherwise. Using the AND rule, the FC decides for $S^* = 1$ only when all the

sensors decide in favor of H_1,

$$\begin{aligned} S^* = 1 : \text{if} \quad & \sum_{i=1}^{n} u_i = n \\ S^* = 0 : \text{otherwise} \quad & \end{aligned} \tag{2.13}$$

while, by applying the OR rule [50], it decides for $S^* = 1$ if any of the nodes decide for H_1. The global decision of the OR rule is given by

$$\begin{aligned} S^* = 1 : \text{if} \quad & \sum_{i=1}^{n} u_i \geq 1 \\ S^* = 0 : \text{otherwise} \quad & \end{aligned} \tag{2.14}$$

The most general voting rule is the k-out-of-n rule wherein the FC decides $S^* = 1$ when at least k nodes out of n decide for H_1. A special case is the majority rule where $k = \lfloor \frac{n}{2} \rfloor$. The k-out-of-n rule is formalized by

$$\begin{aligned} S^* = 1 : \text{if} \quad & \sum_{i=1}^{n} u_i \geq k \\ S^* = 0 : \text{otherwise} \quad & \end{aligned} \tag{2.15}$$

The performance of the fusion rule are evaluated by $Q_D = P(S^* = 1|H_1)$ and $Q_{FA} = P(S^* = 1|H_0)$ as the global probabilities of detection and false alarm. By assuming that the each sensor makes its decision independently of the others, the expressions for the performance are given below. $Q_{D_{AND}}$ and $Q_{FA_{AND}}$ obtained by applying the AND rule are given by

$$\begin{aligned} Q_{D_{AND}} &= \prod_{i=1}^{n} P_{d_i}, \\ Q_{FA_{AND}} &= \prod_{i=1}^{n} P_{fa_i}, \end{aligned} \tag{2.16}$$

while, for the OR rule, $Q_{D_{OR}}$ and $Q_{FA_{OR}}$ are given by

$$\begin{aligned} Q_{D_{OR}} &= 1 - \prod_{i=1}^{n} (1 - P_{d_i}), \\ Q_{FA_{OR}} &= 1 - \prod_{i=1}^{n} (1 - P_{fa_i}), \end{aligned} \tag{2.17}$$

and finally, for the k-out-of-n rule

$$\begin{aligned} Q_{D_{kn}} &= \sum_{i=k}^{n} \binom{n}{i} P_{d_i}^{i} (1 - P_{d_i})^{(n-i)} \\ Q_{FA_{kn}} &= \sum_{i=k}^{n} \binom{n}{i} P_{fa_i}^{i} (1 - P_{fa_i})^{(n-i)}. \end{aligned} \tag{2.18}$$

A comparative performance evaluation of the three voting rules under different settings is conducted in [51].

When there are no limitations on the bandwidth, the overall performance of the fusion technique can be improved by sending more detailed information to the FC [52]. This information can be a statistics or the LRT value about the system (known as "soft decision"). We present three simple and common information fusion rules: Square Law Combining (SLC) [53], Maximal Ratio Combining (MRC) and Selection Combining (SC) [54]. MRC is an optimal combination scheme when the Channel Side Information (CSI) is known at the FC [55]. The CSI is the Signal to Noise ratio (SNR) between the sensor i and the system and it is denoted by γ_i. This information is not required by SC and SLC. By applying the SLC, the FC sums all the received data as follows

$$U_{SLC} = \sum_{i=1}^{n} u_i, \tag{2.19}$$

MRC is a modified version of SLC wherein a weight w_i proportional to SNR is assigned to each information provided by the sensors as follows

$$U_{MRC} = \sum_{i=1}^{n} w_i u_i. \tag{2.20}$$

On the other hand, the SC selects the information of the sensor experiencing the maximum SNR

$$U_{SC} = \max_{\gamma_i}(u_1, u_2, \ldots, u_n). \tag{2.21}$$

Using soft combination rules, the decision for the value of S^* is made by comparing the combined information to a threshold ζ. In [56], [57], [58], [59], it is shown that at the cost of higher overhead and channel quality requirement, the soft fusion rules provide better performance than hard fusion rules.

2.3.2 Advanced Fusion Rules

Many forms of advanced information fusion schemes exist. They depend on many factors ranging from the a-priori available information to the application scenarios wherein these schemes are applied. Examples of advanced information fusion techniques include evidential belief reasoning, fusion and fuzzy reasoning, rough set fusion for imperfect data, random set theoretic fusion and others [60]. Here we consider statistical information fusion due to its perfect match with distributed sensor network applications. In addition, statistical information fusion has a very rich background and in the literature it is the most common approach applied to distributed sensor networks [45], [61]. By adopting such an approach, the LRT has to be performed at the FC after receiving the information from the sensors. Given the vector $\mathbf{u} = u_1, u_2, \ldots, u_n$ with the information sent to the FC, the Bayesian information fusion that minimizes the average cost of the global decision is given by

$$\Lambda(\mathbf{u}) = \frac{p(u_1, u_2, \ldots, u_n|H_1)}{p(u_1, u_2, \ldots, u_n|H_0)} \underset{S^*=0}{\overset{S^*=1}{\gtrless}} \frac{P(H_0)(C_{10} - C_{00})}{P(H_1)(C_{01} - C_{11})} = \lambda \tag{2.22}$$

where, C_{ij} is the cost of the global decision of the FC in favor of H_i when the system is in state H_j. Hereafter, given the conditional independence assumption of local *decisions* provided by the sensors, the Bayesian fusion becomes

$$\begin{aligned} \Lambda(\mathbf{u}) &= \frac{p(u_1, u_2, \ldots, u_n|H_1)}{p(u_1, u_2, \ldots, u_n|H_0)} \\ &= \prod_{i=1}^{n} \frac{p(u_i|H_1)}{p(u_i|H_0)} \\ &= \prod_{S_1} \frac{p(u_i = 1|H_1)}{p(u_i = 1|H_0)} \prod_{S_0} \frac{p(u_i = 0|H_1)}{p(u_i = 0|H_0)} \\ &= \prod_{S_1} \frac{1 - P_{md_i}}{P_{fa_i}} \prod_{S_0} \frac{P_{md_i}}{1 - P_{fa_i}}. \end{aligned} \tag{2.23}$$

where, S_j is the set of the sensors for which $u_i = j$. By applying the logarithm to the last part of Equation (2.23), we obtain

$$\sum_{S_1} \log\left(\frac{1 - P_{md_i}}{P_{fa_i}}\right) + \sum_{S_0} \log\left(\frac{P_{md_i}}{1 - P_{fa_i}}\right) \underset{S^*=0}{\overset{S^*=1}{\gtrless}} \log(\lambda) \tag{2.24}$$

which also can be expressed as

$$\sum_{i=1}^{n} \left[u_i \log\left(\frac{1-P_{md_i}}{P_{fa_i}}\right) + (1-u_i)\log\left(\frac{P_{md_i}}{1-P_{fa_i}}\right)\right] \underset{S^*=0}{\overset{S^*=1}{\gtrless}} \log(\lambda) \tag{2.25}$$

This is an optimal fusion rule and it is known as the Chair-Varshney rule [62]. It can be seen as performing a weighted sum over the local decisions provided by the sensors. The Chair-Varshney rule requires the knowledge of the local performances of the sensors , the a-priori probabilities about the system state, and the costs.

These limitations can be overcome by using the Neyman-Pearson rule [63] maximizing the detection probability at the FC (Q_D) while constraining the false alarm probability Q_{FA}.

$$\Lambda(\mathbf{u}) = \frac{p(u_1, u_2, \ldots, u_n|H_1)}{p(u_1, u_2, \ldots, u_n|H_0)} \underset{S^*=0}{\overset{S^*=1}{\gtrless}} \lambda. \tag{2.26}$$

The threshold λ is set in such a way to satisfy the constraint on the global false alarm probability Q_{FA}, in the following

$$\sum_{\Lambda(\mathbf{u})>\lambda} P(\Lambda(\mathbf{u})|H_0) = Q_{FA}. \tag{2.27}$$

If the FC must take a decision as soon as the information sent by the sensors is enough, the sequential probability ratio test (SPRT) can be applied globally. In this case, if M information samples are already enough, the FC can make the global decision. If not, the FC can collect more information from the sensors. The SPRT can be formalized as follows

$$\prod_{i=1}^{M} \frac{p(u_i|H_1)}{p(u_i|H_0)} = \Lambda(\mathbf{u}), \tag{2.28}$$

then, the decision is made according to the rule:

$$\begin{cases} \Lambda(\mathbf{u}) \geq \Upsilon_1, & S^* = 1 \\ \Upsilon_0 < \Lambda(\mathbf{u}) < \Upsilon_1 & \text{take new value} \\ \Lambda(\mathbf{u}) \leq \Upsilon_0, & S^* = 0 \end{cases} \tag{2.29}$$

The thresholds Υ_0 and Υ_1 are computed by constraining both $Q_{FA} = \alpha_{FC}$ and $Q_{MD} = \beta_{FC}$ as in the following equation:

$$\begin{aligned} \Upsilon_1 &\approx \frac{1-\beta_{FC}}{\alpha_{FC}} \\ \Upsilon_0 &\approx \frac{\beta_{FC}}{1-\alpha_{FC}}. \end{aligned} \tag{2.30}$$

2.4 Game Theory in a Nutshell

As anticipated in the introduction, game theory is a mathematical discipline that studies the situations of competition and/or cooperation, between decision makers known as players. Game theoretic concepts apply whenever the actions of several decision-makers are mutually dependent, that is their choices mutually affect each other. For this reason, game theory is sometimes referred to as *interactive decision theory*.

Although examples of games occurred long before, the birth of modern Game Theory as a unique field was in 1944, with the book "Theory of Games and Economic Behavior" by John von Neumann and Oskar Morgenstern [64]. Game Theory provides tools to formulate, model and study strategic scenarios in a wide variety of application fields ranging from economics and political science to computer science. A fundamental assumption in almost all variants of Game Theory is that each decision maker is rational and intelligent. A rational player is one who has certain specific preferences over the outcomes of the game. A player intelligence is its ability to always select the move that gives him the most preferable outcome, given his expectation about his opponents move. The objective of game-theory is to predict how rational players will play the game, or, to give advice on the strategies to be followed when playing the game against rational opponents.

Game-theoretic models are highly abstract representations of classes of real life situations for which satisfying solutions for players are recommended with desirable properties. Game Theory encompasses a great variety of situations depending on the number of players, the way the players interact, the knowledge that a player has on the strategies adopted by the opponents, the deterministic or probabilistic nature of the game, etc. In all the models, the basic entity is the player, which should be interpreted as an individual

or as a group of individuals, making a decision or following a strategy. A distinction can be made between situations in which the players have common goals and hence play a cooperative game and situations in which the players have different and possibly conflicting goals. In the latter case we say that the game is non-cooperative. Hybrid games contain cooperative and non-cooperative players. For instance, coalitions of players are formed in a cooperative game, but they play in a non-cooperative manner. Another possible classification between game-theoretical models concerns the amount of information available to the players about each other, leading to Games with Perfect or Imperfect Information. A more common classification is made between simultaneous and sequential games. Simultaneous games are games where both players move simultaneously, or if they do not move simultaneously, they are unaware of the earlier players' moves so that their moves are effectively simultaneous. On the contrary, in sequential games the players have some knowledge about earlier moves. This knowledge does not need to be complete i.e., a player may know that an earlier player did not perform one particular move, while he does not know which of the other available moves the first player actually chose.

Game representations are used to differentiate between simultaneous and sequential games. Here, we introduce briefly games in normal form. A normal form game is represented by a matrix where, for the 2-players case, one player is considered as the *row* player, and the other as the *column* player. Each row or column represents a strategy (which is the move selected by the player) and each entry in the matrix represents the payoff, that is the final outcome of the game for each player for every combination of strategies. An example of a 2-player game in normal form is shown in Table 2.2.

	Strategy 1	Strategy 2
Strategy 1	(a, b)	(c, d)
Strategy 2	(e, f)	(j, h)

Table 2.2: *Example of game representation in normal form. The row player is player 1 and the column player is player 2. The entries of the table are the payoffs of the game for each pair of strategies.*

For non-cooperative games, the normal form is used when the players choose their move or set of moves once and for all at the beginning, that is when all the players' decisions are made simultaneously [65].

In this book, we consider only game in normal form (also called strategic form), which is the basic game model studied in non-cooperative game theory. A game in normal form lists each player strategies, and the outcomes that result from each possible combination of choices. A two-player normal form game is defined by the four-tuple $G(\mathcal{S}_1, \mathcal{S}_2,$
$v_1, v_2)$, where $\mathcal{S}_1 = \{s_{1,1} \dots s_{1,n_1}\}$ and $\mathcal{S}_2 = \{s_{2,1} \dots s_{2,n_2}\}$ are the sets of strategies available to the first and second player, and $v_l(s_{1,i}, s_{2,j}), l = 1, 2$ is the payoff (also called utility) of the game for the l^{th} player, when the first player chooses the strategy $s_{1,i}$ and the second chooses $s_{2,j}$. A profile is a pair of strategies $(s_{1,i}, s_{2,j})$. Games in normal form are compactly represented by matrices, namely, payoff matrices.

2.4.0.1 Nash Equilibirium

Given a game, determining the best strategy that each player should follow to maximize its payoff is not easy. Even more, a profile which is optimum for both players may not exist. A common goal in Game Theory is to determine the existence of equilibrium points, i.e., profiles that, to a certain extent, represent a satisfactory choice for both players. While there are many definitions of equilibrium, the most famous and commonly adopted is the one by John Nash [66, 67]. In a 2-player game, a profile (s_{1,i^*}, s_{2,j^*}) is a Nash equilibrium if:

$$\begin{aligned} v_1(s_{1,i^*}, s_{2,j^*}) &\geq v_1(s_{1,i}, s_{2,j^*}) \quad \forall s_{1,i} \in \mathcal{S}_1 \\ v_2(s_{1,i^*}, s_{2,j^*}) &\geq v_2(s_{1,i^*}, s_{2,j}) \quad \forall s_{2,j} \in \mathcal{S}_2, \end{aligned} \tag{2.31}$$

where for a zero-sum game $v_2 = -v_1$. In practice, a profile is a Nash equilibrium if none of the players can improve its payoff by changing its strategy unilaterally. This profile is known as the *equilibrium point* or *saddle point* of the game. Two types of Nash equilibria exist: pure strategy Nash equilibrium and mixed strategy Nash equilibrium.

Pure Strategy Nash Equilibirum:
A pure strategy Nash equilibrium is a Nash equilibrium in which each player selects a single strategy and plays it. Then, a pure strategy profile (s_{1,i^*}, s_{2,j^*}) is a Nash equilibrium for the game with s_{1,i^*} and s_{2,j^*} being the *pure* strategies for player 1 and player 2, respectively. This means that none of the player will improve his payoff by changing his strategy unilaterally.

Mixed Strategy Nash Equilibirum:
Players could also follow another, more complicated strategy. They can randomize their choice over the set of available strategies according to a certain probability distribution. Such a strategy is called a *mixed strategy*. Given a 2-players game $G(\mathcal{S}_1, \mathcal{S}_2, v_1, v_2)$, let $\Pi(\mathcal{Z})$ be the set of all the probability distributions over the set $\mathcal{Z} = \{z_1, \ldots, z_n\}$. Then, the *set of mixed strategies* for a player i are all probability distributions over its strategy set $\mathcal{S}_i$, namely, $\Pi(\mathcal{S}_i)$.

The set of *mixed strategy profiles* is the cartesian product of single mixed strategy sets $\mathbf{\Pi} = \Pi(\mathcal{S}_1) \times \Pi(\mathcal{S}_2)$. The mixed strategy of the player i over the set of $\mathcal{S}_i$ is denoted as $p_i(\mathcal{S}_i)$ and for simplicity, it is used as p_i. So, if player 1 selects a mixed strategy p_1 over the set of strategies $\mathcal{S}_1$ and has beliefs about the second player mixed strategy $p_1 \in \Pi(\mathcal{S}_2)$ over $\mathcal{S}_2$ then, the expected payoff for player 1 given the mixed strategy profile (p_1, p_2) is the weighted average of player's 1 expected payoffs to his pure strategies s_1, where the weight associated to each pure strategy is the probability assigned to that strategy by the player's mixed strategy. This can be expressed as:

$$\begin{aligned} v_1(p_1, p_2) &= \sum_{s_{1,i} \in \mathcal{S}_1} p_1(s_{1,i}) v_1(s_{1,i}, p_2) \\ &= \sum_{s_{1,i} \in \mathcal{S}_1} p_1(s_{1,i}) \left(\sum_{s_{2,j} \in \mathcal{S}_2} p_2(s_{2,j}) \cdot v_1(s_{1,i}, s_{2,j}) \right). \end{aligned} \tag{2.32}$$

It is known that every normal form game with a finite sets of moves has at least one Nash equilibrium in mixed strategies [66].

For strictly competitive games, Nash equilibrium has interesting properties. Let G be a two-players zero-sum game and (s_{1,i^*}, s_{2,j^*}) be a Nash

equilibrium; then, s_{1,i^*} maximizes the first player payoff in the worst case scenario, i.e., assuming that second player selects his most profitable strategy corresponding to the most harmful move for the first player. Similarly, s_{2,j^*} maximizes the second player payoff. We also have that [67]

$$\max_{s_{1,i}\in\mathcal{S}_1}\min_{s_{2,j}\in\mathcal{S}_2} v_1(s_{1,i},s_{2,j}) = \min_{s_{2,j}\in\mathcal{S}_2}\max_{s_{1,i}\in\mathcal{S}_1} v_1(s_{1,i},s_{2,j}) = v_1(s_{1,i^*},s_{2,j^*}) \quad (2.33)$$

As a consequence of relation (2.33), if many equilibrium points exist, they all yield the same payoff. In a 2-player game, a player minmax value is always equal to its maxmin value, and both are equal to the Nash equilibirum value as shown by Von Neumann's Minimax Theorem [68].

Equation (2.33) can be equivalently solved by solving two separate Linear Programming (LP) [69] problems, one for each player that moves first (corresponding to the outer max/min). Since the problems are *duals*, it turns out that only one LP has to be solved to find the optimal strategies for the players [70, 71].

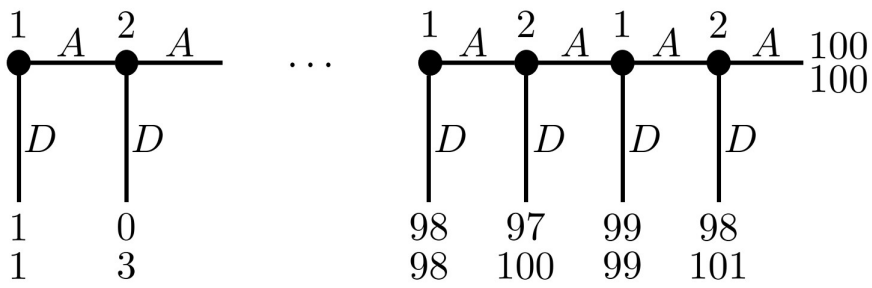


Figure 2.5: *Centipede game example.*

For completeness, it should be mentioned that in some types of games, Nash equilibria can be highly inefficient. An example of this situation is the Centipede game [67]. This game is an extensive-form game in which two players alternately get a chance to take a larger portion of a continuously increasing amount of money. As soon as a player decides to take the money,

the game ends with that player getting the larger amount while the other player gets less amount of money. An example of the centipede game is shown in Figure 2.5. In this figure, a 1 at a black circle (decision node) denotes a decision opportunity for player 1 and a 2 at a decision node denotes a decision opportunity for player 2. The top number at the end of each vertical line is the payoff for the first player and the bottom number is the payoff for the second player. The first player has the first move: if he chooses to take the money by playing D, both players will get a payoff of 1, otherwise, the opportunity to make a decision passes to player 2. In the second step, the second player has the decision opportunity: if he chooses D, player 1 gets payoff of 0 and player 2 gets 3, otherwise, by playing A the opportunity to make a decision passes to player 1, and so on till the end of the game tree. If both players always choose to not take the money (by playing A), they both receive payoff of 100 at the end of the game. Therefore, both players will receive a payoff of 100 if they always choose A rather than D and will receive payoff of 1 if player 1 chooses D on his first move. Using backward induction – which is the process of reasoning backward from the end of a problem – game theory predicts that the first player will choose to play D at the very first move and both players will receive a payoff of 1.

However, in experimental studies as the one shown in [72], the authors found that theoretical predictions of game-theory is rarely followed. By considering several versions of the game with different game length, they found that in only 7% of the four-move games, 1 % of the six-move games, and 15 % of the high payoff games did the first player choose to play D in the first move. This contradiction can be explained by two reasons. The first is that some people are selfless and prefer to cooperate with the other player by always playing A, rather than taking down the money. Another reason is that people may be incapable of making the deductive reasoning necessary to make the rational choice predicted by the Nash equilibrium. The fact that few people play D on the very first move is not surprising, given the small size of the starting payoff when compared with the increasing payoffs as the game progresses. As a result, this example shows a case where the prediction of the Nash equilibrium is not efficient and may contradict real experiments.

2.4.0.2 Dominance Solvable Games

Despite its popularity, the practical meaning of Nash equilibrium is often unclear, since it cannot be guaranteed that the players will end up playing at the Nash equilibrium. A particular kind of normal form games for which stronger forms of equilibrium exist are called dominance solvable games [67]. This concept is directly related to the notion of dominant and dominated strategies. A strategy is said to be *strictly dominant* for one player if it is the best strategy for the player no matter how the other player decides to play. Another form of dominant strategy is the *weak dominant* strategy, which, regardless of what any other player do, gives a payoff at least as high as any other strategy available in the strategy set, and, gives a strictly higher payoff for some profile of other players' strategies. Obviously, if a strictly dominant strategy exists for one of the players, he will surely adopt it. Similarly, a strategy $s_{l,i}$ is *strictly dominated* by strategy $s_{l,j}$, if the payoff achieved by player l choosing $s_{l,i}$ is always lower than that obtained by playing $s_{l,j}$ regardless of the choice made by the other player. Formally, in the 2-players case, a strategy $s_{1,i}$ is *strictly dominated* by strategy $s_{1,k}$ for a player, for instance, player 1, if

$$v_1(s_{1,k}, s_{2,j}) > v_1(s_{1,i}, s_{2,j}) \quad \forall s_{2,j} \in \mathcal{S}_2. \tag{2.34}$$

Alternatively, a strategy $s_{1,i}$ is *weakly dominated* by strategy $s_{1,k}$ for a player, for instance, player 1, if

$$v_1(s_{1,k}, s_{2,j}) \geq v_1(s_{1,i}, s_{2,j}) \quad \forall s_{2,j} \in \mathcal{S}_2. \tag{2.35}$$

Following this definition, a *strictly dominant* strategy is a strategy which strictly dominates all the other strategies in the strategy set.

A possible technique to solve a game is by recursive elimination of the dominated strategies since, all the strategies that a player definitely should not adopt can be removed from the game. In recursive elimination, first, all the dominated strategies are removed from the set of available strategies, since no rational player would ever play them. In this way, a new, possibly smaller game is obtained. Then, at this point, some strategies, that were not dominated before, may be dominated in the remaining game, and hence are eliminated. The process is repeated until no dominated strategy exists for

any player. A *rationalizable equilibrium* is any profile which remains after the recursive elimination of dominated strategies [73, 74]. If at the end of the process only one profile is left, the remaining profile is said to be the *only rationalizable equilibrium* of the game, which is also the only Nash equilibrium point. A dominance solvable game is a game that can be solved according to the procedure described above. Note that the removal of weakly dominated strategies will possibly cause the loss of some of the Nash equilibria of the game. For instance, consider the payoff matrix in Table 2.3, strategy T is weakly dominant for player 1 and strategy L is strictly dominant for player 2, and (T,L) is a Nash equilibrium, but also (B,L) is a Nash equilibrium that is lost deleting the weakly dominated strategy B.

	L	R
T	$(1,1)$	$(2,0)$
B	$(1,2)$	$(0,0)$

Table 2.3: *Example of removal of weakly dominated strategies will cause the loss of some Nash equilibria. The row player is player 1 and the column player is player 2.*

It goes without saying that the concept of rationalizable equilibrium is a stronger notion than that of Nash equilibrium [75]. In fact, under the assumption of rational and intelligent players, it can be seen that the players will choose the strategies corresponding to the unique rationalizable equilibrium since it will maximize their payoffs. An interesting, related notion of equilibrium is that of dominant equilibrium. A *dominant equilibrium* is a profile that corresponds to dominant strategies for both players and is the strongest kind of equilibrium that a game in normal form may have.

2.5 Conclusion

We introduced some basic notions of detection theory and outlined some detection techniques used locally at the sensors as well as the corresponding decision strategy. Specifically, we discussed the Bayesian detector, the Neyman-Pearson detector and the SPRT detector used locally at the sensors and their

extension to be deployed globally at the FC. Then, we listed some common information fusion techniques that can be employed by the FC in the centralized setup. Specifically, we discussed the AND, OR and k-out-of-n rules when sensors send hard decisions, and the SLC, MRC and SC combination rules when sensors send soft decisions.

Additionally, we gave a brief introduction to game theory. First, we introduced games in normal form and we explained some solution concepts to these games, namely, Nash equilibrium and dominance solvability. Then, we discussed some examples of games in normal form to clarify the mechanics of the games and their solutions. As game theory may be used to model competitions, in this book, it will play a fundamental role in modeling the competition between the players, namely, the adversaries as the attackers and the sensor network. Throughout the book we will focus on 2-player games in normal form in which the first player (the attackers) and the second player (the defender) aim at achieving opposite objectives. At one hand, the attackers want to introduce decision errors in the distributed sensor network in order to achieve some selfish or malicious objectives. On the other hand, the defender aims at protecting the sensor network against attackers and provide the most possible robust detection and decision performance. By adopting such a model, we are aiming at finding possible equilibria describing the interplay between the attackers and the defender and then, try to find out who will win the game.

Chapter 3

Security Attacks and Defenses in Distributed Sensor Networks

3.1 Introduction

Information fusion in distributed sensor networks may suffer from various threats and attacks. An incentive for the adversary could be the critical nature of the phenomenon observed by sensors, such as troops passage in a battlefield, traffic flow [76], [77], [30], image authenticity in front of a court for crime witness [5], [78], and many others. In addition, deceiving the decision of the network about the phenomenon could be beneficial for the attacker e.g. to gain exclusive access to the spectrum in cognitive radio networks [79], change an item reputation in online reputation systems [80] and others.

A fundamental and key enabling factor to ensure the proper functionality of sensor networks is securing them against attacks and threats. A fundamental step for the protection of sensor networks is securing the information fusion process in order to ensure *trusted and accurate* decisions about the observed phenomenon. To do so, the information fusion process should take into account the possible presence of sensors under the control of an adversary, knowing that they can have various malicious objectives.

In this chapter, we outline the most common attacks in centralized distributed sensor networks as well as in consensus-based decentralized networks. In addition, we present the most common countermeasures to protect the network from these attacks.

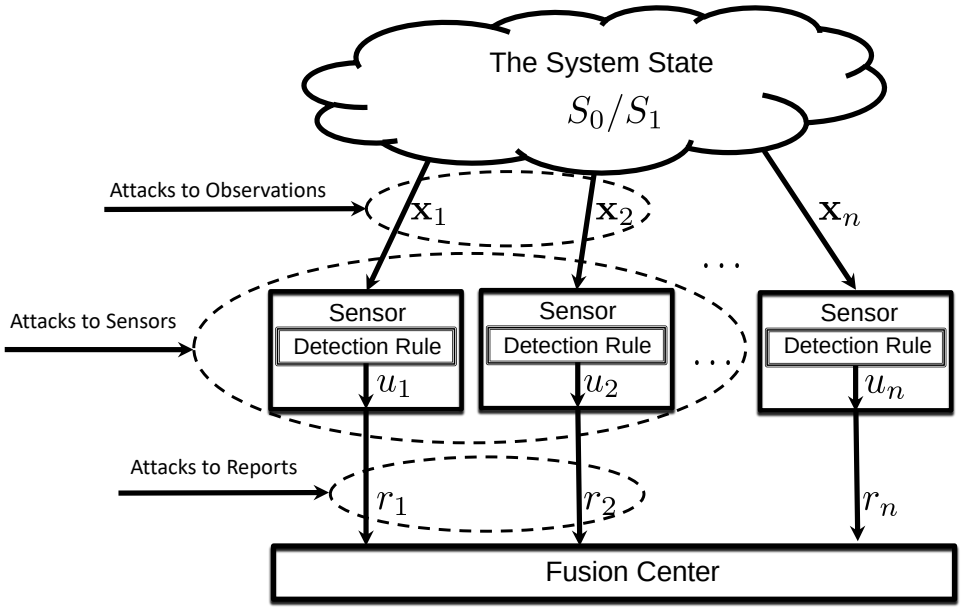


Figure 3.1: *Classification of attacks to distributed sensor networks.*

3.2 Attacks to Distributed Sensor Networks

We start by considering the centralized network setup illustrated in Figure 3.1. The illustration shows several possible adversarial setups. In all these setups, the information fusion process is carried out at the level of the binary decisions $r_1, \ldots, r_n$ provided by the sensors. The adversary can carry out its attacks in three positions, which are:

- *The observations about the phenomenon used by the sensors to make the local decision.* In this attack, the adversary can access and eavesdrop the observations and modify them since he has control over the observation channel between the system and the sensor network. In

this way, the adversary can control what the sensors will observe about the phenomenon and consequently, deceive the local decision and hence, indirectly, corrupt the information fusion process performed later at the FC.

- *The sensors themselves.* In this attack, a fraction (or all) of the sensors are under the control of the adversary. Then, the adversary can modify the detection and decision rules, the decision thresholds or the data computed locally to be sent to the FC. Altering the information computed locally by the sensors can maliciously affect the result of the fusion process since a part of the information fused is unilaterally altered by the adversary.

- *The information sent by the sensors.* In this case, the attackers does not have control over the sensors but instead, they have access to the links between the sensors and the FC. Alike the case of attacking the sensors, this attack can make the information fusion fail. The difference between the two types of attacks is that in the latter, the adversary does not have access to the observations.

3.2.1 Attacks to the Observations

The n sensors in the network observe the phenomenon through the vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. In this scenario, a fraction of the links, let us say $\alpha = b/n$, is under the control of the adversary. The variable α, could be either the exact fraction of the links under the control of the adversary, or the probability that a link is under the control of the adversary. The adversary can modify the vectors from $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_b$ to $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_b$ so to induce a wrong local decision at the sensors $1, \ldots, b$. The objective of the adversary could vary from pushing the b sensors to change their decision from H_0 to H_1 and viceversa or only corrupting the functionality of the network by inducing random errors. Consequently, a fraction of the information sent to the FC will be incorrect and this will affect the global decision at the FC. This setup is very general and can be used to model a variety of situations. Here, we present two examples of such an adversarial setup: the jammer attacks in wireless communication networks [81] and the Primary User Emulation Attack (PUEA) in cognitive

radio networks [82].

3.2.1.1 Jammer Attack

Due to the open and shared nature of the wireless medium, together with the advancement of wireless technologies and software, wireless networks can be easily monitored, accessed and broadcast on by a transmitter. The adversary can observe the communication links and the information between wireless entities, and launch Denial of Service (DoS) attacks by injecting wrong information messages. Severe types of DoS attacks can be launched in wireless networks which can block the wireless medium and prevent other wireless devices from even communicating with each other. These attackers are known as *jammers* and continuously transmit radio frequency signals to occupy the channel and then block the information flow in the network or between the network and the system [77]. Therefore, a *jammer* is an adversary who intentionally tries to interfere with the transmissions and receptions of wireless communications.

The objective of a jammer is to interfere with legitimate wireless communications. It can achieve this goal by either preventing the transmission of the information from the source, by preventing the reception of the information, or by modifying the information exchanged between the transmitter and the the receiver. Many types of jammers exist depending on their objective and behavior, we may have:

- A Constant jammer continuously transmitting signals to block the communication between some selected network entities.
- A Deceptive jammer constantly injecting regular information into the links with no separation between subsequent transmissions. In this way, the receiver will be deceived into believing that there is an information coming and will remain in receiving mode.
- A Random jammer who randomly alternates between sleeping and attacking modes in order to save energy.
- Reactive jammer: the jammer stays quiet when the channel is idle and there is no information exchange and starts transmitting as soon as it

senses any activity on the channel. Therefore, a reactive jammer targets the reception of a message. The advantage of this kind of jammer is that it is harder to detect.

A complete survey of jammer attacks and their feasibility in wireless networks can be found in [81].

The connection between jammer attacks and attacking the observations is straightforward as the jammer can:

- Block the system state information observed by the sensors to let them believe that the activity does not exist which means that the system is in state S_0 while it can be in S_1.
- Inject a specific information value on the link to change the sensor local decision from H_0 to H_1 or viceversa, like the case of a *deceptive jammer*.

3.2.1.2 Primary User Emulation Attack

In Cognitive Radio Networks, guaranteeing a trustworthy spectrum sensing is a particularly important problem. In this setup, the Secondary Users (SU) try to opportunistically use the channel owned by the Primary User (PU) when it is idle. The main concern in spectrum sensing is the ability to distinguish between the PU and SU signals. To do so, an SU should continuously scan the spectrum for the presence of PU signals in the candidate bands. If an SU detects a PU signal in the current band, it must immediately switch to another band. On the other hand, if the SU detects the presence of another SU, it runs a coexistence mechanism [83,84] to share spectrum bands. Distinguishing the two types of signals is not trivial, especially in hostile environments. Many techniques are proposed to increase the accuracy of PU activity detection including: Energy detectors, Cyclostationary detectors, Wave-based detectors, Matched filter detectors and so on [85,86]. The most common approach to improve spectrum sensing accuracy is *collaborative spectrum sensing* [50, 59], in which, the SUs collaboratively scan the spectrum and send their results, to a FC that makes a final decision about spectrum occupancy. This collaboration among SUs can be also implemented in a decentralized fashion [83].

A Primary User Emulation Attacker (PUEA) [79, 87, 88] is an adversary that modifies the air interface of a CR to mimic the characteristics of a PU

signal, thereby causing the SUs to mistakenly detect the adversary signal as a PU signal. The high reconfigurability of software-defined CR devices makes PUEAs possible and realistic [89].

When the attacker detects no PU activity, it sends *jamming signals* emulating PU's activity, so to let the SUs believe that a PU is active and hence, prevent them from using the available spectrum. This attacker can be seen as a new type of DoS jamming attack specific to cognitive radio networks scenario [88].

Based on the objective of the adversary, PUEAs can be classified into two classes: Selfish PUEA and Malicious PUEA. A selfish PUEA aims at increasing the usage of the spectrum by the attacker. By finding an available spectrum band and preventing other SUs from accessing that band, the adversary can gain alone the access to the spectrum band. On the other hand, a malicious PUEA aims at impeding spectrum sharing by deceiving the spectrum sensing proces. In this way, SUs always detect the presence of a PU and move to another band [90].

3.2.2 Attacks to the Sensors

In the case of attacks to the sensors, the FC has to tackle with the presence of a number of malevolent sensors, which deliberately alter their information reports to induce a global decision error. According to a consolidated literature, such nodes are referred to as *byzantine nodes* or simply *Byzantines* [32, 33]. Note that a byzantine sensor can decide to alter its report by relying on its observations of the system state $S_i, i \in \{0, 1\}$, but usually it does not have access to the observations available to the other sensors and their information reports. In this setup, a fraction α of the n sensors is under the control of the attacker which, in order to make the fusion process fail, alter the local information or decision prior to sending them to the FC. From the perspective of the FC, the problem can be viewed as a robust distributed information fusion problem as the information from the sensors is a mixture of good and adversarial data.

3.2.2.1 Spectrum Sensing Data Falsification Attacks

In a cognitive radio network setup, Spectrum Sensing Data Falsification (SSDF) [86], [35, 36, 91–94] refers to a SU that sends altered local spectrum sensing results, which will possibly result in erroneous decisions by the FC. The SSDF attack is an example of a *Byzantine* attack targeting the spectrum sensing process. In cognitive radio networks, a spectrum sensing failure can be caused by malfunctioning SUs or SSDF attacks. A malfunctioning SU is unable to produce reliable local information and may send wrong sensing information to the FC. On the other hand, in SSDF attacks, a malicious SU intentionally sends falsified reports to the FC in the attempt to cause a failure in the information fusion process. It is shown in [95] that, under certain assumptions, even a single byzantine sensor can make the information fusion process fail.

Depending on the attack objective and behavior, SSFD attacks can be classified into the following categories:

- Malicious SUs [96] send false sensing results so to confuse the FC about spectrum occupancy. The objective of malicious SSDF attack is to lead the FC or the rest of the SUs to decide the absence of a PU signal when it is present, or make them believe that there is a PU signal when there is not. Consequently, in the first case, the SUs will refrain from using the specific band, while in the second case they will cause harmful interference to PU.

- Greedy SUs [97] continuously report that a specific spectrum band is occupied by a PU. This can be seen as a selfish attack aiming at occupying the available band alone by forcing the other SUs to evacuate it.

- Unintentionally misbehaving SUs [79] send wrong information reports about PU activity in the spectrum, not because they are attackers, but because of a problem in their software or hardware such as random faults or virus [98–100].

3.2.3 Attacks to the Reports

In this case, the adversary can access the links between the sensors and the FC. This may correspond to a situation in which the adversary does not control the nodes but only the communication link between the nodes and the fusion center, or to the case of byzantine sensors which, for some reasons, cannot observe the information data at the input of the sensor, or decide to not exploit such a knowledge.

3.3 Defenses Against Attacks to Distributed Sensor Networks

Having presented the most common adversarial setups and attacks in distributed sensor networks, we now describe the most common countermeasures and show how they contribute to protect the network. In general, these countermeasures can directly modify the information fusion process or introduce a pre-step before the fusion process so to filter out the adversary effect before performing the fusion.

3.3.1 Defenses against Attacks to the Observations

3.3.1.1 Defenses Against Jammer Attack

For the jammer attack, the typical defenses involve the usage of spread-spectrum communication such as frequency hopping or code spreading [30]. Frequency-hopping spread spectrum (FHSS) [101] is a method of transmitting signals by rapidly changing the carrier frequency among many available channels using a pseudo-random sequence known at both the transmitter and the receiver. The lack of knowledge of the frequency selection by the attacker makes jamming the frequency being used not possible. However, since the range of possible frequencies is limited, a jammer may instead jam a wide set of the frequencies increasing its possibility to succeed in the attack.

Code spreading [101] is another way to defend against jamming attacks. A pseudo-random spreading sequence is used to multiplex the signals for transmission. This sequence is known to both the transmitter and the receiver and

without it the attacker cannot jam the communication channel. This method is widely used in mobile networks [102]. However, code spreading has high design complexity and energy consumption, thus limiting its usage in energy limited scenarios like wireless sensor networks.

3.3.1.2 Defenses Against PUEA

In its report, the FCC [103] states that: "No modification to the incumbent signal should be required to accommodate opportunistic use of the spectrum by SUs". This restriction should be followed when designing security mechanisms to defend against PUEA or any other attack specific to the cognitive radio setup.

For PUEA, few defense mechanisms assume that the location of the PU is known. Those mechanisms are called location-based defense mechanisms. We follow this classification of the defense mechanisms which has been introduced in [79].

Location-based defense mechanisms against PUEA In [82], the authors utilize two pieces of information to develop their defense protocol: the location of the PU transmitter and the Received Signal Strength (RSS). The RSS information is collected by a separate wireless sensor network. The defense scheme consists of three phases: first, it verifies if the signal characteristics are similar to those of the PU or not, then it tests the received signal energy based on the location information, and last, it tests the localization of the signal transmitter. A transmitter who does not pass any of these three phases will be considered as PUEA and will be omitted. The drawbacks of this method are: first, the location information about the PU is not always available, especially in small networks, and second, the RSS may have large fluctuations even within small area networks.

In [104], Fenton's approximation and Sequential Probability Ratio Test (SPRT) are used to analytically model the received power at the SUs. The SUs compare the received power in a band of interest to a threshold. If the power is below the threshold the band is considered to be free. On the other hand, if the band is tagged as occupied, the SUs test whether the detected

signal source is a legitimate PU or a PUEA. Based on the assumption that the attackers and the SUs are uniformly distributed, two statistical formulations are proposed to model the Probability Density Function (PDF) of the received power at the SU from a legitimate PU, and the PDF of the received power at the SU from malicious users. Then, the defense mechanism at the SUs tests the two PDFs using an SPRT by performing a binary hypothesis test between H_0 (the signal comes from legitimate PU), and H_1 according to which the signal comes from a PUEA. In this setup, several malicious users can be present in the network and the authors show that when the attackers are too close to the SUs, the false alarm and missed detection probabilities are maximized because the total received power from all the attackers is larger than the received power from the legitimate PU. A drawback of this work is the possibility of an endless loop of the SPRT that leads to very long sensing times. This work is extended in [89] where the authors use Neyman Pearson composite hypothesis testing to solve the endless loop problem of SPRT.

Other defense proposals based on localization algorithms use the Time of Arrival (TOA), Time Difference of Arrival (TDOA), and Angle of Arrival (AOA) in order to distinguish between a PU legitimate signal and PUEA [84]. In TOA, SUs receive signals that contain the receiver location and time information. Based on this information, the node can calculate its own position and estimates the PU position. TDOA [105] is a passive localization technique that uses the difference between the arrival times of signals transmitted by a PU but does not know the signal transmission time. TDOA measures the time differences at several receivers with known locations and computes a location estimate of the PU that permits to distinguish between a legitimate and a malicious behavior. In the AOA technique, an SU measures the angle of arrival of the signal from two or more other SUs. If the locations of the other SUs are known, the receiver can compute its own location using triangulation [106]. By using the same method, the AOA information at multiple SUs is used to determine the PU location and hence, help to distinguish between the legitimate PU and PUEA. All of these techniques fail when the PUEA is too close to the PU and, hence, knowing the PU location gives no benefit.

location-agnostic defenses In [107], the authors use the channel impulse response, referred to as "link signature", to determine whether a PU transmitter changes its location or not. They propose the use of a "helper node" located in a fixed position very close to the PU. This node communicates with SUs to help them to verify the PU signals. The SUs do so by verifying the cryptographic link signatures carried out by the helper node which communicates with SUs only when there is no PU transmission. For this reason, the helper node has to sense the PU transmissions and also to differentiate PU signals from PUEA signals. The helper node authenticates the legitimate PU using the first and the second multipath components of the received PU signal at its interface. Then, the helper node compares the ratio of the multipath components to a threshold, and if the ratio is above the threshold, it decides that the signal belongs to a legitimate PU, otherwise that it is a PUEA. Now, the SUs verify the PU transmission by computing the distance between the link signatures of the received signals and those sent by the helper node. If the distance is lower than a threshold, the received signal belongs to a legitimate PU, otherwise, it is a PUEA and it will be discarded.

Other proposals to defend against PUEA contradict with the FCC requirement since they try to modify the PU signal. Part of these proposals modify the PU signal to integrate into it a cryptographic signatures that permits the SU to distinguish the PUs from the PUEAs, like in [108], and other proposed authentication mechanisms between the PU transmitter and the SUs [107].

3.3.2 Defenses against Attacks to Sensors

We now present some countermeasures to defend against the *byzantine* attacks and the Spectrum Sensing Data Falsification (SSDF) attack in cognitive radio networks.

As mentioned earlier in Chapter 2, various information types can be provided by the sensors at different levels of abstraction. In this section, we consider the simplest case in which the sensors send binary decisions about the observed phenomenon, namely, a bit 0 under H_0 and 1 otherwise. We consider this case since it is the most relevant for the rest of the book.

In the absence of Byzantines, the Bayesian optimal fusion rule has been derived in [45, 62] and it is known as Chair-Varshney rule. If the local error

probabilities (P_{MD}, P_{FA}) are symmetric and equal across the sensor network, Chair-Varshney rule boils down to a simple majority-based decision.

In the presence of Byzantines, Chair-Varshney rule requires the knowledge of Byzantines' positions in the binary vector submitted to the FC along with the flipping probability P_{mal} [1]. Since this information is rarely available, the FC may resort to a suboptimal fusion strategy.

An overview of the literature about distributed detection and estimation in the presence of the Byzantines is given in [33]. The authors use the concept of *critical power* of Byzantines (α_{blind}) that was originally introduced in [109] in order to characterize the fraction of Byzantines that makes the decision at the FC no better than flipping a coin. In [110], by adopting a Neyman-Pearson setup and assuming that the byzantine nodes know the true state of the system, the asymptotic performance - as a function of the network size n - obtainable by the FC is analyzed as a function of the percentage of Byzantines in the network. By formalizing the attack problem as the minimization of the Kullback-Leibler Distance (KLD) [111] between the information reports received by the FC under the two hypotheses H_0 and H_1, the blinding percentage, that is, the percentage of Byzantines in the network that makes the FC blind, is determined and shown to be - at least asymptotically - always equal to 50%. This means that unless more than half of the sensors are *Byzantines*, asymptotically, the FC can provide reliable detection and decision.

By observing the system state over a longer observation window, the FC improves the estimation of the sequence of system states by gathering a number of reports provided by the sensors before making a global decision. In cooperative spectrum sensing, for instance, this corresponds to collectively decide about the vacant spectrum bands over a time window, or, more realistically, at different frequency slots. The advantage of deciding over a sequence of states rather than on each single state separately, is that in such a way it is possible for the FC to understand which are the byzantine nodes and discard the corresponding observations (such an operation is usually referred to as *byzantine isolation*). Such a strategy is adopted in [112], where the analysis of [110] is extended to a situation in which the Byzantines do not know

[1] The flipping probability is the probability that the attacker flips its binary local decision about the system state before sending it to the FC.

the true state of the system. Byzantine isolation is achieved by counting the mismatches between the reports received from each sensor and the global decision made by the FC. In order to cope with the lack of knowledge about the strategy adopted by the Byzantines, the decision fusion problem is casted into a game-theoretic formulation, where each party makes the best choice without knowing the strategy adopted by the other party.

A slightly different approach is adopted in [113]. By assuming that the FC is able to derive the statistics of the reports submitted by honest sensors, byzantine isolation is carried out whenever the reports received from a node deviate from the expected statistics. In this way, a correct decision can be made also when the percentage of Byzantines exceeds 50%. The limit of this approach is that it does not work when the reports sent by the Byzantines have the same statistics of those transmitted by the honest nodes. This is the case, for instance, in a perfectly symmetric setup with equiprobable system states, symmetric local error probabilities, and an attack strategy consisting of simple decision flipping.

3.3.2.1 Defenses to SSDF in Cognitive Radio Networks

The byzantine attack in a cognitive radio context is known as SSDF attack. In this part, we present the most common countermeasures against these attacks. In the scenario considered here, the group of SUs send *binary* decisions about PU activity to the FC which, in turn, decides about the spectrum occupancy by fusing the decisions using an information fusion rule. In some of these works, when the SUs are not trusted a priori, a trust or reputation metric is assigned to each SU in the network depending on its behavior.

In [95] the proposed scheme calculates *trust values* for SUs based on their past reports. This metric can become unstable if attackers are not present or if there are not enough reports. For this reason, the authors also compute a *consistency value* for each SU. If the consistency value and the trust value fall below certain thresholds, the SU is identified as a Byzantine and its reports are not considered in the fusion rule. The authors evaluate the proposed scheme using two fusion rules, namely, the OR rule and the 2-out-of-n rule. A drawback of this work is that only *one adversary* is considered in the evaluation.

The authors of [114] use a *reputation metric* to detect and isolate attackers from honest SUs. This metric is computed by comparing the report of each SU to the final decision made at the FC. The metric increments by one if the report and the final decision mismatch (more reliable SUs have low metric values). If the reputation metric of an SU exceeds a predefined threshold, its reports are isolated and not used in the fusion rule. By adopting majority voting as the fusion rule, the authors show that when the percentage of attackers in the network is below 40%, the probability of isolating the attackers can exceed 95%, while the isolation probability of honest SUs is very close to zero. This defense scheme is similar to [95] with the difference that here the authors do not restore the reputation metric if an SU is temporary misbehaving and thus, [95] is considered to be a more fair approach.

Weighted Sequential Probability Ratio Test (WSPRT) as a modified version of the SPRT test is proposed in [34] to assign a weight w_i to each SU in the network as follows:

$$\Lambda(\mathbf{u}) = \prod_{i=1}^{n} \Big(\frac{P(u_i|H_1)}{P(u_i|H_0)} \Big)^{w_i}. \tag{3.1}$$

In Equation (3.1), the FC computes the product of the likelihood ratios for each decision provided by the SUs. Based on the likelihood value for each SU report, the FC assigns to it a weight $w_i \in [0, 1]$ that determines its contribution in the final decision made at the FC. The weight is computed based on a reputation rating r_i assigned to each SU. If the report of the SU matches with the final decision at the FC, its reputation rating r_i is increased by one, otherwise it is decreased. Then, using a non-decreasing function $f(.)$, the reputation rating is mapped to a weight w_i to be used for each SU report in the WSPRT. In addition, the reputation rating of a misbehaving SU can be restored to zero just after a few instants if it starts behaving correctly again. Each SU implements an SPRT and decides for H_1 (PU presence) or H_0 (PU absence), by comparing the local spectrum sensing measurement with two predefined thresholds, wich are computed by constraning the false alarm and missed detection probabilities as explained in Chapter 2, Section 2.2. If the output falls between these thresholds, no decision is made and the SU takes a new sample. For the simulation, the authors assume two constant strategies for the adversary, namely, the "always true" SSDF attackers, which always

reports a vacant spectrum, and the "always false" SSDF attackers, always reporting the spectrum as occupied. Furthermore, eight different information fusion rules are considered at the FC: AND, OR, Majority, SPRT, WSPRT and LRT with three different thresholds. For the "always-false" attack case, the simulation results show that for all fusion rules, except the OR and AND rules, the correct detection ratio decreases as the number of the attackers increases. For the other two rules, the correct sensing ratio does not significantly change, but it is lower than the other cases. For the "always-true" case, the results show that the performance of the majority rule decreases significantly as the number of attackers increase, which means that this rule is more vulnerable to this type of attack.

In [115], the authors propose an attack called the "hit-and-run" attack. By knowing the fusion technique used by the FC, this attacker alternates between honest and lying modes. The attacker estimates its own *suspicious level* and as long as it is below a threshold h, it reports falsified decisions. If the suspicious level falls below a threshold, it switches to honest mode. On the defender side, when the suspicious level of an SU becomes larger than h, a point is assigned to this user. When the cumulative points exceed a predefined threshold, the reports of this SU are ignored permanently. Simulations show that the scheme achieves good performance for up to three attackers. A drawback of this method is that a user is permanently removed from a CRN if it collects enough points then, some honest but temporarily misbehaving SUs can be permanently removed from the network. In addition, the authors assume a non-realistic scenario in which the adversary knows the reports of the other SUs.

In [116], a Double-Sided Neighbor Distance (DSND) algorithm is used for the detection of the attackers. An SU is characterized as an adversary if its reports to the FC are too far or too close to the reports sent by other SUs. Two attack models are considered: the independent attack, where an adversary does not know the reports of the honest SUs, and the dependent attack, where the adversary is aware of the reports of the others. The results show that, for the independent attack, the adversary can always be detected when the number of spectrum sensing iterations tends to infinity. For the dependent attack, the adversary can avoid been detected if it has accurate

information about the missed detection and false alarm probabilities and then follows them in his attacking strategy.

3.3.3 Defenses against Attacks to Reports

As discussed above, in this attack the adversary can access the links between the sensors and the FC. From the FC point of view, both types of attacks force it to consider that a part of the received information is malevolently altered and consider this fact when fusing the information. Therefore, the defense methods and algorithms presented in Section 3.3.2 can also be applied in this case.

3.4 Conclusion

In this chapter we have presented the most common attacks and countermeasures in different adversarial setups. We considered the centralized version of the problem and reviewed the attacks and defenses in several adversarial settings. In addition, we provided real-life examples of attacks and some specific mitigations for each adversarial setup.

In the rest of the book, we will focus on information fusion in distributed sensor network in which some of the sensors are *Byzantines*. In addition to that, we will specifically consider the case in which the sensors report binary decisions to the FC.

Chapter 4

Adversarial Decision Fusion: A Heuristic Approach

4.1 Introduction

In this chapter, we present a heuristic approach for adversarial decision fusion in centralized distributed sensor networks. In the considered setup, the fusion center is required to make a decision about the status of an observed system by relying on the information provided by the nodes. Decision fusion must be carried out in an adversarial setting, by taking into account the possibility that some of the nodes are Byzantines and they malevolently alter their reports to induce a decision error. We assume that the byzantine nodes do not know the true state of the system and act by flipping the local decisions with a certain probability P_{mal}. The fusion center first tries to understand which are the byzantine nodes and then makes a decision by discarding the suspect nodes. Despite the simplicity of the setup, this case contains all the ingredients of more complex situations, hence its presentation and analysis is very instructive and already provides interesting insights into the achievable performance of decision fusion in distributed sensor networks under adversarial conditions.

With the above ideas in mind, the goal of this chapter is twofold. First of all, it introduces a soft identification strategy whereby the fusion center can isolate the byzantine nodes from the honest ones. Then, it introduces a game-theoretic formulation of the decision fusion problem with attacked nodes, thus providing a rigorous framework to evaluate the performance achievable by the fusion center and the Byzantines, when they both play at the equilibrium. The game-theoretic approach is used to compare the fusion strategy described in this chapter with the one presented in [112]. Finally, the performance of the soft identification scheme are evaluated by means of numerical simulations.

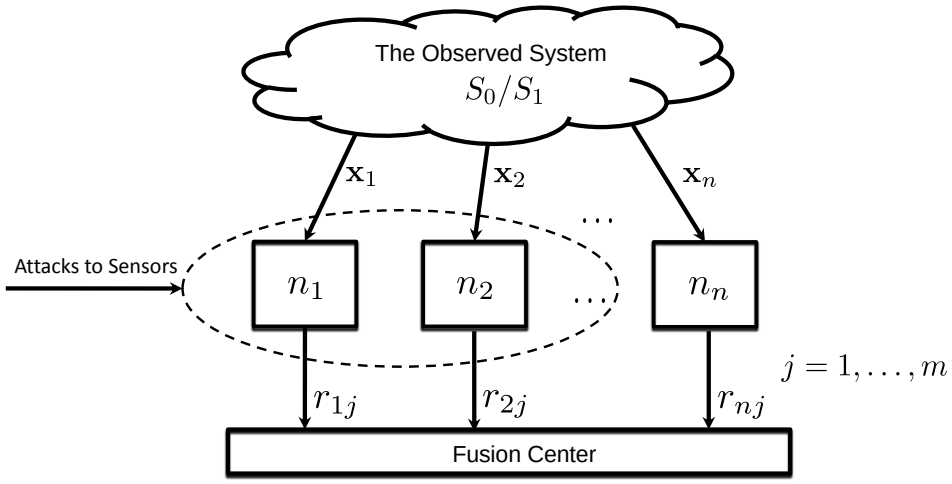


Figure 4.1: *Heuristic setup for decision fusion under adversarial conditions.*

A graphical representation of the setup presented in this chapter is given in Figure 4.1. Following the notation introduced in Chapter 2 the n nodes of the distributed sensor network observe a system through the vectors $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_n$. Based on such vectors, the nodes compute n reports, say $\mathbf{r}_1, \mathbf{r}_2 \ldots \mathbf{r}_n$ and send them to a fusion center.[1] The fusion center gathers all the reports and makes a final decision about the state of the observed system. The system can be only in two states S_0 and S_1. Additionally, the reports correspond to local decisions on the system status made by the nodes, i.e. the reports are binary values and $r_i \in \{0, 1\}$ for all i.

[1]Notation $\mathbf{r}_i$ is introduced to denote the general report received by the fusion center from a node i, that can be either honest or byzantine.

4.2 Decision Fusion with Isolation of Byzantines

4.2.1 Problem formulation

In the scenario described above, the Fusion Center (FC) uses the *byzantine isolation* strategy described in chapter 3. In the following an exact formulation of such an approach is given.

As we anticipated, we consider the case of binary reports. Specifically, each node makes a local decision about the state of the observed system and forwards its one-bit decision to FC, which must decide between hypothesis H_0 and hypothesis H_1. A fixed fraction α of the n nodes (or links between the nodes and the FC) is under the control of byzantine attackers that, in order to induce an error in the fusion process, corrupt the reports by flipping the one-bit local decisions with probability P_{mal}. Under this assumption, the probability that a node is a Byzantine depends weakly on the state of the other nodes when the network size is large enough. By referring to Figure 4.1, the above attack corresponds to the insertion of a binary symmetric channel with crossover probability P_{mal} in the attacked links.

The strategy adopted by the fusion center consists in trying to identify the attacked nodes and remove the corresponding reports from the fusion process. To do so, the FC observes the decisions taken by the nodes over a time period m, and makes the final decision on the state of the system at each instant j only at the end of the observation period. To further elaborate, for each instant j, we indicate the reports received from the nodes as $r_j^n = (r_{1j}, r_{2j}, ..., r_{nj})$ where $r_{ij} \in \{0, 1\}$. The fusion center applies an l-out-of-n fusion rule to r_j^n (that is, the fusion center decides in favor of H_1 if l out n nodes decided for such a hypothesis) to make an intermediate decision on the status of the system at time j. Let us indicate such a decision by $d_{int}(j)$. The local decisions made by the i-th node over the time window m, are denoted as $\mathbf{u}_i = (u_{i1}, u_{i2}, \ldots, u_{im})$ where u_{ij} is the local decision at instant j, $j \in \{1, \ldots, m\}$. The relationship between u_{ij} and the status of the system at time j is ruled by the following equations, which take into account the probability of a decision error:

$$P\left(u_{ij} = 1 | H_1\right) = P_{d_i} \tag{4.1}$$

$$P\left(u_{ij}=1|H_0\right)=P_{fa_i}, \tag{4.2}$$

where P_{d_i} and P_{fa_i} are, respectively, the probability of correct detection and false alarm for node i. In the following, we assume that the states assumed by the system over subsequent instants are independent of each other. Errors at different nodes and different times are also assumed to be independent.

By assuming that transmission takes place over error-free channels, for honest nodes we have $r_{ij}=u_{ij}$, while for the byzantine nodes we have $r_{ij}\neq u_{ij}$ with probability P_{mal}. Then, for the byzantine reports we have:

$$P(r_{ij}=1|H_1)=P_{\text{mal}}(1-P_{d_i})+(1-P_{\text{mal}})P_{d_i}, \tag{4.3}$$

$$P(r_{ij}=1|H_0)=P_{\text{mal}}(1-P_{fa_i})+(1-P_{\text{mal}})P_{fa_i}. \tag{4.4}$$

Given the observation vector r_j^n for each j $(j=1,..,m)$, in order to remove the fake reports from the data fusion process, the FC proceeds as follows: it associates to each node i a *reputation score* Γ_i, based on the consistency of the reports received from that node with the intermediate decisions $d_{int}(j)$ over the entire time window m. Then, the FC isolates the nodes whose reputation is lower than a threshold η and decides about the system state by fusing only the remaining reports.

4.2.2 Byzantine Identification: hard reputation measure

In this section, we describe a simple way to build a hard reputation measure. For each node i, the FC computes a reputation score by counting the number of times that the reports received from that node are different from the intermediate decisions $d_{int}(j)$ during the observation window m. The *hard reputation score* $\Gamma_{H,i}$ is hence defined as $\Gamma_{H,i}=\sum_{j=1}^{m}\mathcal{I}(r_{ij}=d_{int}(j))$ where $\mathcal{I}(\cdot)$ is the indicator function, which is equal to 1 when the argument of $\mathcal{I}$ is true, and 0 otherwise. Accordingly, the nodes whose reputation is lower than a threshold η are removed from the fusion process. For each j, the final decision is taken by relying on an l'-out-of-n' rule, where l' is the final decision threshold and n' is the number of nodes remaining after that the thought-to-be byzantine nodes have been discarded.

In [112], the above scheme is shown to be able to mitigate the effect of byzantine attacks when $\alpha<0.5$, a situation in which the Byzantines are not

able to blind the FC by attacking the network independently, which is the only case considered in this chapter.

4.3 Decision Fusion with Soft Identification of Malicious Nodes

In this section, we present an isolation strategy which removes the Byzantines from the network according to a soft[2] reliability measure. For any instant j and given the vector r_j^n with the reports, the new isolation strategy relies on the estimation of the following probabilities:

$$\begin{aligned} &P\left(u_{ij}=1, r_j^n\right), \\ &P\left(u_{ij}=0, r_j^n\right). \end{aligned} \tag{4.5}$$

The rationale behind this strategy relies on the observation that when r_j^n gives a clear indication regarding the intermediate decision about the state, e.g., almost all reports are 0, then for all nodes k reporting 0, we have $P\left(u_{kj}=0, r_j^n\right) >> P\left(u_{kj}=1, r_j^n\right)$, and hence the reputation is high, as desired. On the contrary, for all nodes k' reporting 1, we have $P\left(u_{k'j}=0, r_j^n\right) \cong P\left(u_{k'j}=1, r_j^n\right)$, and hence the reputation is low. Instead, when r_j^n does not give a clear indication regarding the state, the reputation will be low in all cases, in contrast to the hard reputation mechanism where the reliability of the intermediate decision is not taken into account.

For this reason, the reputation score of a node is measured as follows. For each j we first compute:

$$R_{ij} = \left| \log \left[\frac{P\left(u_{ij}=0, r_j^n\right)}{P\left(u_{ij}=1, r_j^n\right)} \right] \right|. \tag{4.6}$$

Then we set:

$$\Gamma_{S,i} = \sum_{j=1}^{m} R_{ij}. \tag{4.7}$$

[2]We point out that the soft identification method is soft with regard to the identification of the Byzantines in the intermediate step of the Byzantines identification process, but is not used in the final decision step as the majority rule is applied.

To evaluate (4.6), we start rewriting the joint probabilities within the log as follows (for notation simplicity, we omit the index j):

$$P\left(u_i, r^n\right) = P\left(r^n|u_i, H_0\right) P\left(u_i, H_0\right) + P\left(r^n|u_i, H_1\right) P\left(u_i, H_1\right). \tag{4.8}$$

To go on, we make the simplifying assumptions that the reports received by the FC from different nodes are conditionally independent, that is, they are independent when conditioning to H_0 or H_1. This is only approximately true since in the presented scenario we operate under a fixed number of Byzantines, and then the probability that a node is Byzantine depends (weakly) on the state of the other nodes. Such dependence decreases when the number of nodes increases. To clarify this assumption, let us denote the probability of a node being Byzantine as $P(B)$. If the nodes are independent of each others, $P(B) = \alpha$ for every node i in the network and in this sense, the meaning of the probability of a node being a Byzantine is equivalent to the fraction of Byzantines in the network. On the other hand, if the probability that a node is a Byzantine depends on the status of the other nodes, this will not be the case anymore. Let us take the vector of nodes $[1, 2, \ldots, n]$ then, the probability that the first node is a Byzantine is $P(B_1) = \alpha = \frac{M}{n}$. Then, we take the second node, by assuming that the first node is Byzantine, the probability that the second is also a Byzantine becomes $P(B_2|B_1) = \frac{M-1}{n-1}$. For large M we have $P(B_2|B_1) \approx \frac{M}{n} = P(B_1)$.

Let us now consider the quantity $P(r_{j'}|u_i, H_0)$. When $i = j'$, we can omit the conditioning to H_0 since r_i depends on the system status only through u_i. On the other hand, when $i \neq j'$, we can omit conditioning to u_i, due to the conditional independence of node reports. A similar observation holds under H_1. Then we can write:

$$\begin{aligned} P\left(u_i, r^n\right) = P\left(r_i|u_i\right) \Big\{ & P\left(u_i|H_0\right) P\left(H_0\right) \prod_{j' \neq i} P\left(r_{j'}|H_0\right) \\ & + P\left(u_i|H_1\right) P\left(H_1\right) \prod_{j' \neq i} P\left(r_{j'}|H_1\right) \Big\}, \end{aligned} \tag{4.9}$$

where, $P\left(r_i|u_i\right) = (1-\alpha) + \alpha(1 - P_{\text{mal}}) = (1 - \alpha P_{\text{mal}})$ if $r_i = u_i$, and $P\left(r_i|u_i\right) = \alpha P_{\text{mal}}$ if $r_i \neq u_i$. Moreover, we have $P(u_{j'} = 1|H_1) = P_{d_{j'}}$ and

$P(u_{j'} = 1|H_0) = P_{fa_{j'}}$. In addition:

$$P\left(r_{j'}|H_0\right) = (1 - \alpha P_{\text{mal}})P\left(u_{j'} = r_{j'}|H_0\right) + \alpha P_{\text{mal}} P\left(u_{j'} \neq r_j|H_0\right), \quad (4.10)$$

$$P\left(r_{j'}|H_1\right) = (1 - \alpha P_{\text{mal}})P\left(u_{j'} = r_{j'}|H_1\right) + \alpha P_{\text{mal}} P\left(u_{j'} \neq r_{j'}|H_1\right). \quad (4.11)$$

By inserting the above expressions in (4.10) and (4.11), we can compute the soft reputation score $\Gamma_{S,i}$. Then, the FC relies on $\Gamma_{S,i}$ to distinguish honest nodes from byzantine ones. Specifically, the distinction is made by isolating those nodes whose reputation score $\Gamma_{S,i}$ is lower than a threshold η (hereafter, we will set $P_{fa_i} = P_{fa}$ and $P_{d_i} = P_d$, $\forall i$).

We conclude, by observing that, strictly speaking, FC is required to know α and the flipping probability P_{mal}. With regard to α, we assume that the FC knows it. As to P_{mal}, in the next sections, we will see that choosing $P_{\text{mal}} = 1$ is always the optimum strategy for the attackers, and hence the FC can assume that $P_{\text{mal}} = 1$. This assumption is reasonable in this case since the amount of information available at the FC about the Byzantines is not large enough to let the FC identify the Byzantines easily and force them to use a lower P_{mal}. Then, it is better for him to adopt a conservative approach.

4.4 A Game-Theoretical Approach to the Decision Fusion Problem

In this section, we evaluate the performance achieved by using the strategy introduced in the previous section and compare it with the hard identification strategy. To do so, we use a game-theoretic approach in such a way to analyze the interplay between the choices made by the attackers and the fusion center.

4.4.1 The Decision Fusion Game

In the scenario presented in this chapter, the FC is given the possibility of setting the local sensor threshold for the hypothesis testing problem at the nodes and the fusion rule, while the Byzantines can set the flipping probability P_{mal}. In addition, the FC is endowed with the possibility of setting the isolation threshold η, as well as the final fusion rule after removal of byzantine

nodes. The performance are evaluated in terms of the overall error probability after the removal step. We suppose that the FC does not act strategically on the local sensor threshold; then P_d and P_{fa} are fixed and known to the FC. With regard to the Byzantines (B), they are free to decide the flipping probability P_{mal}.

With the above ideas in mind, we define the general decision fusion game as follows:

Definition 1. *The $DF(\mathcal{S}_{FC}, \mathcal{S}_B, v)$ game is a zero-sum strategic game, played by the FC and B, defined by the following strategies and payoff.*

- *The set of strategies available to the FC is given by all the possible isolation thresholds η, and the values of l and l' in the l-out-of-n intermediate and final decision rules:*

$$\mathcal{S}_{FC} = \{(l, \eta, l');\ l, l' = 1, .., n,\ \eta_{\min} \leq \eta \leq \eta_{\max}\}, \tag{4.12}$$

  where $\eta_{\min}$ and $\eta_{\max}$ depend on the adopted isolation scheme.

- *The set of strategies for B are all the possible flipping probabilities:*

$$\mathcal{S}_B = \{P_{\text{mal}},\ 0 \leq P_{\text{mal}} \leq 1\}. \tag{4.13}$$

- *The payoff v is the final error probability after malicious node removal, namely $P_{e,ar}$. Of course, the FC wants to minimize $P_{e,ar}$, while B tries to maximize it.*

Applying the above definition to the identification schemes introduced so far, we see that for the case of hard reputation measure (DF_H game), the values of the isolation threshold η range in the set of integers from 0 to m, while for the scheme based on the soft removal of the malicious nodes (DF_S game) η may take all the continuous values between $\eta_{\min} = \min_{i=1,..,n} R_{ij}$ and $\eta_{\max} = \max_{i=1,..,n} R_{ij}$.

4.4.2 Equilibrium Point of the Decision Fusion Game

With regard to the optimum choice for the Byzantines, several works either conjecture or demonstrate (in particular cases) that $P_{\text{mal}} = 1$ is a dominant

strategy [112, 117]. Even in the case considered here, simulations confirm that $P_{\text{mal}} = 1$ is indeed a dominant strategy for both the hard and the soft identification schemes. This means that, notwithstanding the introduction of an identification scheme for discarding the reports of malicious nodes from the fusion process, the optimum for the Byzantines is always flipping the local decisions before transmitting them to the FC. This means that for the Byzantines it is better to use all their power ($P_{\text{mal}} = 1$) in order to make the intermediate decision fail than to use a lower P_{mal} to avoid being identified. As a consequence of the existence of a dominant strategy for B, the optimum strategy for the FC is the triple (l^*, η^*, l'^*) which minimizes $P_{e,ar}$ when $P_{\text{mal}} = 1$. By exploiting a result derived in [45] for the classical decision fusion problem, the optimal value l^* determining the intermediate fusion rule is given by

$$l^* = \frac{\ln\left[(P(H_0)/P(H_1))\{(1-p_{10})/(1-p_{11})\}^n\right]}{\ln\left[\{p_{11}(1-p_{10})\}/\{p_{10}(1-p_{11})\}\right]}, \tag{4.14}$$

where $p_{10} = p(r = 1|H_0)$ and $p_{11} = p(r = 1|H_1)$, evaluated for $P_{\text{mal}} = 1$. With regard to η and l', we have:

$$(\eta^*, l'^*) = \arg\min_{(\eta, l')} P_{e,ar}((l^*, \eta, l'), P_{\text{mal}} = 1). \tag{4.15}$$

Depending on the adopted isolation scheme, we have a different expression for $P_{e,ar}$ and then different η^*'s and l'^*'s as well. The minimization problem in (4.15) is solved numerically for both hard and soft isolation in the next section. According to the previous analysis, $((l^*, \eta^*, l'^*), P^*_{\text{mal}})$ is the only *rationalizable equilibrium* for the DF game, thus ensuring that any rational player will surely choose these strategies. The value of $P_{e,ar}$ at the equilibrium represents the achievable performance for the FC and is used to compare the effectiveness of data fusion based on soft and hard Byzantine isolation.

4.5 Performance Analysis

We now evaluate the performance at the equilibrium for the two games DF_H and DF_S, showing that the soft strategy outperforms the hard one, in terms of $P_{e,ar}$. We also give a comparison of the two schemes in terms of isolation error probability.

η_H / P_{mal}	0.6	0.7	0.8	0.9	1
4	0.0016	0.0087	0.0354	0.1109	0.2746
3	0.0015	0.0078	0.0262	0.06628	**0.1982**
2	0.0016	0.0080	0.0281	0.0726	0.1998
1	0.0016	0.0087	0.0354	0.1109	0.2746
0	0.0016	0.0087	0.0354	0.1109	0.2746

Table 4.1: *Payoff of the DF_H game for $\alpha = 0.46$ and $P_d = 0.8$, $P_{fa} = 0.2$.*

In all the simulations, we consider a sensor network with $n = 100$ nodes. We assume that the probability of the two states S_0 and S_1 are the same. We run the experiments under the following settings: $P_d = 1 - P_{fa}$ takes values in the set $\{0.8, 0.9\}$ and $\alpha \in \{0.4, 0.41, 0.42, ..., 0.49\}$, corresponding to a number of honest nodes ranging from 51 to 60. The observation window m is set to 4. For each setting, the probability of error $P_{e,ar}$ of the two schemes is estimated over 50000 simulations.

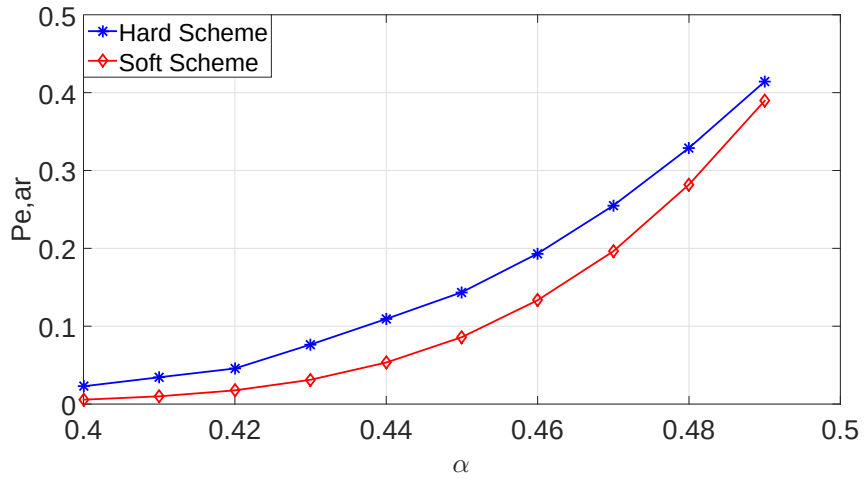
Due to the symmetry of the experimental setup with respect the two states, we have that $p_{10} = p_{01} = 1 - p_{11}$. Accordingly, from (4.14) we get that $l^* = n/2$ and then the majority rule is optimal for any P_{mal} (not only at the equilibrium). Besides, still as a consequence of the symmetric setup, the optimality of the majority rule is experimentally proved also for the final fusion rule, regardless of the values of η and P_{mal}. Then, in order to ease the graphical representation of the game, we fix $l^* = 50$ and $l'^* = n'/2$ and remove these parameters from the strategies available to the FC. The set of strategies η and P_{mal} of the two players, FC and Byzantines, are quantized to get finite sets of strategies and represent the game in normal form (a finer quantization is used for η in the case of soft isolation).

Tables 4.1 and 4.2 show the payoff matrix for the DF_H and DF_S games when $\alpha = 0.46$ and $P_d = 0.8$ (very similar results are obtained for different values of these parameters). For the DF_S game, the threshold values are obtained from the reliability interval $[\eta_{S,\min}, \eta_{S,\max}]$. Since the reliability measures take different values for different P_{mal}, a large number of thresholds have been considered, however for sake of brevity, we show the results obtained with a rather coarse quantization interval, especially far from the equilibrium point.

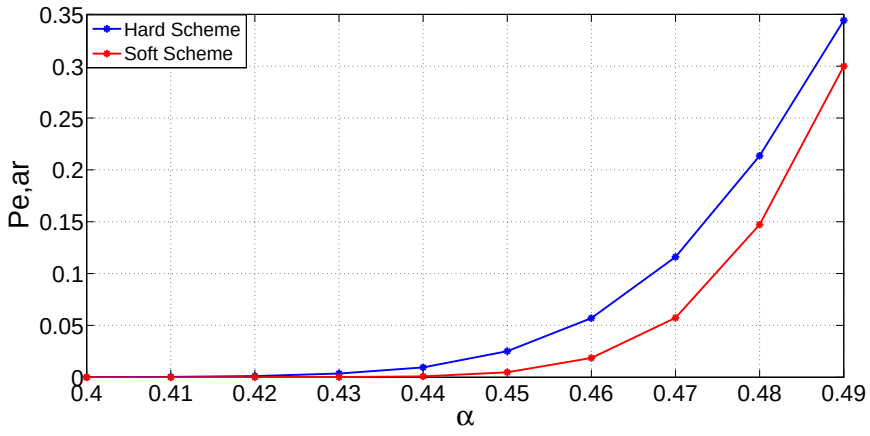
η_S / P_{mal}	0.6	0.7	0.8	0.9	1
$\eta_{S,\text{min}}$	0.0009	0.0035	0.0131	0.0596	0.2253
·	0.0009	0.0035	0.0131	0.0596	0.1889
·	0.0009	0.0035	0.0131	0.0596	0.1589
·	0.0009	0.0035	0.0131	0.0596	0.1401
·	0.0009	0.0035	0.0131	0.0596	0.1405
·	0.0009	0.0035	0.0131	0.0596	**0.1375**
·	0.0009	0.0035	0.0131	0.0596	0.1528
·	0.0009	0.0035	0.0131	0.0596	0.1801
·	0.0009	0.0035	0.0131	0.0596	0.2192
·	0.0009	0.0035	0.0131	0.0596	0.2742
·	0.0009	0.0035	0.0131	0.0361	0.2742
·	0.0009	0.0035	0.0131	0.0209	0.2742
·	0.0009	0.0035	0.0131	0.0586	0.2742
·	0.0009	0.0035	0.0131	0.1108	0.2742
·	0.0009	0.0035	0.0088	0.1108	0.2742
·	0.0009	0.0035	0.0054	0.1108	0.2742
·	0.0008	0.0021	0.0355	0.1108	0.2742
$\eta_{S,\text{max}}$	0.0006	0.0011	0.0355	0.1108	0.2742

Table 4.2: *Payoff of the DF_S game for $\alpha = 0.46$ and $P_d = 0.8$, $P_{fa} = 0.2$.*

As to the strategy of the Byzantines, the simulation results confirm the dominance of $P_{\text{mal}} = 1$ for both games. Looking at the performance at the equilibrium, we see that the DF_S game is more favorable to the FC, with a $P_{e,ar}$ at the equilibrium equal to 0.1375 against 0.1982 for the DF_H game. In Figure 4.2, the two games are compared by plotting the corresponding payoffs at the equilibrium for various values of α in the interval $[0.4, 0.49]$. Upon inspection of the figure, the superiority of the soft isolation scheme is confirmed. Finally, we compare the two schemes in terms of capability of isolation of the byzantine nodes. The ROC curve with the probability of correct isolation of byzantines nodes (P_{ISO}^{B}) versus the erroneous isolation of honest nodes (P_{ISO}^{H}), obtained by varying η, is depicted in Figure 4.3 for both schemes. The curves correspond to the case of $\alpha = 0.46$ and $P_d = 0.8$. As we can see, the soft isolation strategy allows to obtain a slight improvement of the isolation performance with respect to isolation based on a hard reputation score.



(a)



(b)

Figure 4.2: *Error probability $P_{e,ar}$ at the equilibrium for $P_d = 0.8$ (a) and $P_d = 0.9$ (b).*

4.6 Conclusions

In this chapter, we presented two defense schemes for decision fusion in the presence of byzantine nodes, relying, respectively, on a hard and soft reputa-

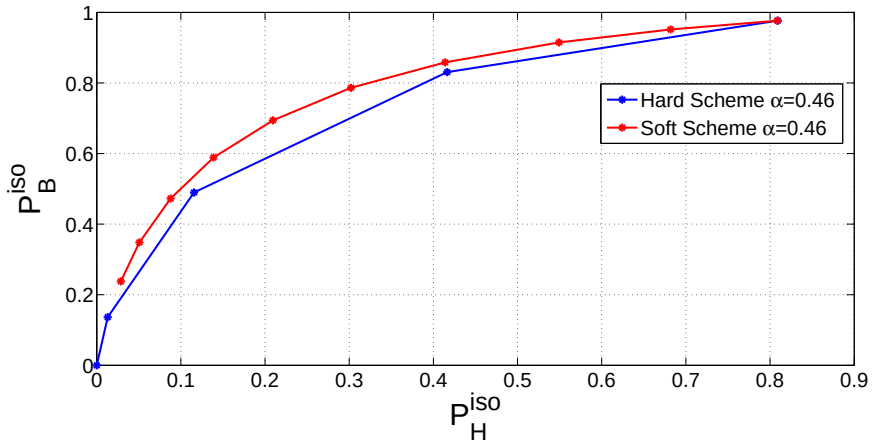


Figure 4.3: P_{iso}^{H} *vs.* P_{iso}^{B} *at* $P_{\text{mal}} = 1.0$, *for* $\alpha = 0.46$ *and* $P_d = 0.8$. *For the soft scheme, 10 thresholds are taken.*

tion measure for the identification of nodes. In order to evaluate the performance of the two schemes, we introduced a game theoretic framework which is particularly suited to analyze the interplay between the fusion center and the Byzantines. We evaluated the equilibrium point of the game by means of simulations and used the payoff at the equilibrium to assess the validity of the two reputation metrics.

Chapter 5

A Game-Theoretic Framework for Optimum Decision Fusion in the Presence of Byzantines

BT: Notation Issue: in Chaopter 6 we use $p()$ instead of $P()$ (then some instances of $p()$ were left - by mistake - in some places in Chaopter 2). Changing notation in Chapter 6 would require to redraw the figures. For simplicity, I suggest **to use everywhere notation $p()$ instead of $P()$ to denote the probability of random variables and of events**.
(@Kassem: note that if we finally switch to the lower case letter, the notation for $P(H_0)$ and $P(H_1)$ should be changed as well in $p(H_0)$ and $p(H_1)$).

5.1 Introduction

This chapter starts from the observation that the knowledge of P_{mal} and the probability distribution of Byzantines across the network introduced in the previous chapter would allow the derivation of the optimum decision fusion rule, thus permitting to the FC to obtain the best achievable performance. It can be also argued that in the presence of such an information discarding the reports received from suspect nodes is not necessarily the optimum strategy, since such reports may still convey some useful information about the status of the system. This is the case, for instance, when $P_{\text{mal}} = 1$. If the FC knows the identity of byzantine nodes, in fact, it only needs to flip the reports received from such nodes to cancel the Byzantines' attack.

By adopting the setup illustrated in Figure 5.1, we first present the derivation of the optimum decision fusion rule when the FC knows both the probability distribution of Byzantines and P_{mal}. The analysis goes along a line which is similar to that used in [62] to derive the Chair-Varshney optimal

fusion rule. As a matter of fact, by knowing P_{mal} and assuming that the probability that a node is Byzantine is fixed and independent on the other nodes, the Chair-Varshney rule can be easily extended to take into account the presence of Byzantines. In contrast to [62], however, the optimal fusion rule derived in this chapter, makes a joint decision on the whole sequence of states hence permitting to improve the decision accuracy. Furthermore, the analysis is not limited to the case of independently distributed Byzantines, but several distributions of the Byzantines across the network are considered. We also describe an efficient implementation of the optimum fusion strategy based on Dynamic Programming [118].

In order to cope with the lack of knowledge regarding P_{mal}, a game-theoretic according to which the FC arbitrarily sets the value of P_{mal} to a guessed value P_{mal}^{FC} and uses such a value within the optimum fusion rule is introduced. At the same time, the Byzantines choose the value of P_{mal} so to maximize the error probability, without knowing the value of P_{mal}^{FC} used by the fusion center. The payoff is defined as the overall error probability, with the FC aiming at minimizing it, while the goal of the Byzantines is to maximize it. With regard to the knowledge that the FC has about the distribution of Byzantines, we present several cases, ranging from a maximum entropy scenario in which the uncertainty about the distribution of Byzantines is maximum, through a more favorable situation in which the FC knows the exact number of Byzantines present in the network. Having defined the game, numerical simulations are used to derive the existence of equilibrium points, which identify the optimum behavior for both the FC and the Byzantines in a game-theoretic sense.

Numerical simulations also help to get more insights into the optimum strategies at the equilibrium and the achievable performance under various settings. The simulations results of the optimum strategy are compared to those in [112] and Chapter 4. Simulation results also confirm the intuition that, in some instances, it is preferable for the Byzantines to minimize the mutual information between the status of the observed system and the reports submitted to the FC, rather than always flipping the decision made by the local nodes. This is especially true when the length of the observed sequence and the available information about the Byzantine distribution allow a good

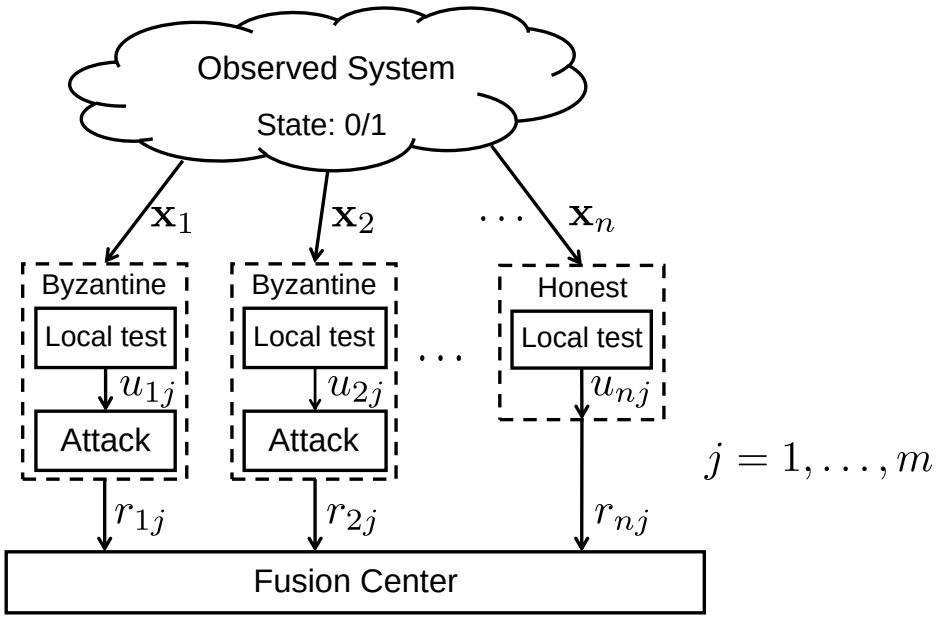


Figure 5.1: *Sketch of the adversarial decision fusion scheme. BT: I do not think that there should be any difference between this figure and the figure in Chapter 4 - that is, Figure 4.1. - ..... Then, I would use this figure in place of Fig 4.1 since this is more clear and detailed (Fig. 4.1 doesn't say much and the notation used in the text is not reported in the figure). At that point, you could refer to the same figure in this chapter as well, without including it again.*

identification of byzantine nodes.

5.2 Optimum fusion rule

In the rest of the book, a different notation is used to distinguish random variables from instantiations; specifically, capital letters are used to denote random variables and lowercase letters for their instantiations. Given a ran-

dom variable X, $P_X(x)$ is used to indicate its probability mass function (pmf). Whenever the random variable the pmf refers to is clear from the context, we will use the notation $P(x)$ as a shorthand for $P_X(x)$.

With the above notation in mind, let $S^m = (S_1, S_2 \dots S_m)$ indicate a sequence of independent and identically distributed (i.i.d.) random variables indicating the state of the system. The independence of the different components of the state vector is a reasonable assumption in several scenarios, e.g. when they represent the status of the frequency spectrum of a cognitive radio system at different frequencies [112]. All states are assumed to be equiprobable, that is $P_{S_j}(0) = P_{S_j}(1) = 0.5$. $U_{ij} \in \{0, 1\}$ is used to denote the local decision made by node i about S_j. Any interaction between the nodes is excluded and U_{ij}'s are assumed to be conditionally independent for a fixed status of the system. This is equivalent to assuming that the local decision errors are i.i.d.

With regard to the position of the Byzantines, let $A^n = (A_1 \dots A_n)$ be a binary random sequence in which $A_i = 0$ (res. $A_i = 1$) if node i is honest (res. byzantine). The probability that the distribution of Byzantines across the nodes is a^n is indicated by $P_{A^n}(a^n)$ or simply $P(a^n)$.

Finally, let $\mathbf{R} = \{R_{ij}\}$, $i = 1 \dots n, j = 1 \dots m$ be a random matrix with all the reports received by the fusion center, accordingly, $\mathbf{r} = \{r_{ij}\}$ denotes a specific instantiation of $\mathbf{R}$. As stated before, $R_{ij} = U_{ij}$ for honest nodes, while $P(r_{ij} \neq u_{ij}) = P_{\text{mal}}$ for byzantine nodes. Byzantine nodes flip the local decisions u_{ij} independently of each other with equal probabilities, so that their action can be modeled as a number of independent binary symmetric channels with crossover probability P_{mal}.

We are now ready to present the derivation of the optimum decision rule on the sequence of states at the FC. We stress that, while considering a joint decision on the sequence of states does not give any advantage in the non-adversarial scenario with i.i.d. states, such an approach permits to improve the accuracy of the decision in the presence of byzantine nodes. Given the received reports $\mathbf{r}$ and by adopting a maximum a posteriori probability criterion, the optimum decision rule minimizing the error probability can be

written as:

$$s^{m,*} = \arg\max_{s^m} P(s^m|\mathbf{r}). \tag{5.1}$$

By applying Bayes rule and exploiting the fact that all state sequences are equiprobable, we obtain:

$$s^{m,*} = \arg\max_{s^m} P(\mathbf{r}|s^m). \tag{5.2}$$

In order to go on, we condition $P(\mathbf{r}|s^m)$ to the knowledge of a^n and then average over all possible a^n:

$$s^{m,*} = \arg\max_{s^m} \sum_{a^n} P(\mathbf{r}|a^n, s^m)P(a^n) \tag{5.3}$$

$$= \arg\max_{s^m} \sum_{a^n} \left(\prod_{i=1}^{n} P(\mathbf{r}_i|a_i, s^m) \right) P(a^n) \tag{5.4}$$

$$= \arg\max_{s^m} \sum_{a^n} \left(\prod_{i=1}^{n} \prod_{j=1}^{m} P(r_{ij}|a_i, s_j) \right) P(a^n), \tag{5.5}$$

where $\mathbf{r}_i$ indicates the i-th row of $\mathbf{r}$. In (5.4) exploited the fact that, given a^n and s^m, the reports sent by the nodes are independent of each other, while (5.5) derives from the observation that each report depends only on the corresponding element of the state sequence. It goes without saying that in the non-adversarial case ($P(a^n) = 1$ for $a^n = (0, \cdots, 0)$ and 0 otherwise) the maximization in (5.5) is equivalent to the following component-wise maximization

$$s_j^* = \arg\max_{s_j} \prod_{i=1}^{n} P(r_{ij}|s_j), \quad \forall j = 1, \cdots, m, \tag{5.6}$$

which corresponds to the Chair-Varshney rule.

We now present the case in which the probability of a local decision error, say ε, is the same regardless of the system status, that is $\varepsilon = Pr(u_{ij} \neq s_j|S_j = s_j)$, $s_j = 0, 1$. For a honest node, such a probability is equal to the probability that the report received by the FC does not correspond to the system status.

This is not the case for byzantine nodes, for which the probability δ that the FC receives a wrong report is

$$\delta = \varepsilon(1 - P_{\text{mal}}) + (1 - \varepsilon)P_{\text{mal}}. \tag{5.7}$$

According to the above setting, the nodes can be modeled as binary symmetric channels, whose input corresponds to the system status and for which the crossover probability is equal to ε for the honest nodes and δ for the Byzantines. With regard to ε, it is reasonable to assume that such a value is known to the fusion center, since it depends on the characteristics of the channel through which the nodes observe the system and the local decision rule adopted by the nodes. The value of δ depends on the value of P_{mal} which is chosen by the Byzantines and then is not generally known to the FC. We will first present the derivation of the optimum fusion rule assuming that P_{mal} is known and then relax this assumption by modeling the problem in a game-theoretic framework.

From (5.5), the optimum decision rule can be written:

$$s^{m,*} = \arg\max_{s^m} \sum_{a^n} \Bigg(\prod_{i:a_i=0} (1-\varepsilon)^{m_{eq}(i)} \varepsilon^{m-m_{eq}(i)} \prod_{i:a_i=1} (1-\delta)^{m_{eq}(i)} \delta^{m-m_{eq}(i)} \Bigg) P(a^n), \tag{5.8}$$

where $m_{eq}(i)$ is the number of j's for which $r_{ij} = s_j$.

As a notice, when there are no Byzantines in the network, the optimum decision in (5.8) boils down to the majority rule.

To go on with the study of the adversarial setup we need to make some assumptions on $P(a^n)$.

5.2.1 Unconstrained maximum entropy distribution

As a worst case scenario, it an be assumed that the FC has no a-priori information about the distribution of Byzantines. This corresponds to maximizing the entropy of A^n, i.e. to assuming that all sequences a^n are equiprobable, $P(a^n) = 1/2^n$. In this case, the random variables A_i are independent of each other and we have $P_{A_i}(0) = P_{A_i}(1) = 1/2$. It is easy to argue that in this

case the Byzantines may impede any meaningful decision at the FC. To see why, let us assume that the Byzantines decide to use $P_{\text{mal}} = 1$. With this choice, the mutual information between the vector state S^m and $\mathbf{R}$ is zero and so any decision made by the FC center would be equivalent to *guessing* the state of the system by flipping a coin. The above observation is consistent with previous works in which it is usually assumed that the probability that a node is Byzantine or the overall fraction of Byzantines is lower than 0.5, since otherwise the Byzantines would always succeed to blind the FC [33].

5.2.2 Constrained maximum entropy distributions

A second possibility consists in maximizing the entropy of A^n subject to a constraint which corresponds to the a-priori information available to the fusions center. We consider two cases. In the first one the FC knows the expected value of the number of Byzantines present in the network, in the second case, the FC knows only an upper bound of the number of Byzantines. In the following, we let N_B indicate the number of Byzantines present in the network.

5.2.2.1 Maximum entropy with given $E[N_B]$

Let $\alpha = E[N_B]/n$ indicate the expected fraction of byzantine nodes in the network. In order to determine the distribution $P(a^n)$ which maximizes $H(A^n)$ subject to α, we observe that $E[N_B] = E[\sum_i A_i] = \sum_i E[A_i] = \sum_i \mu_{A_i}$, where μ_{A_i} indicates the expected value of A_i. In order to determine the maximum entropy distribution constrained to $E[N_B] = \alpha n$, we need to solve the following problem:

$$\max_{P(a^n):\sum_i \mu_{A_i} = n\alpha} H(A^n). \tag{5.9}$$

We now show that the solution to the above maximization problem is obtained by letting the A_i's to be i.i.d. random variables with $\mu_{A_i} = \alpha$. We have:

$$H(A^n) \leq \sum_i H(A_i) = \sum_i h(\mu_{A_i}), \tag{5.10}$$

where $h(\mu_{A_i})$ denotes the binary entropy function[1] and where the last equality derives from the observation that for a binary random variable A, $\mu_A = P_A(1)$. It can be also observed that equality holds if and only if the random variables A_i's are independent. To maximize the rightmost term in Equation (5.10) subject to $\sum_i \mu_{A_i} = n\alpha$, we observe that the binary entropy is a concave function [111], and hence the maximum of the sum is obtained when all μ_{A_i}'s are equal, that is when $\mu_{A_i} = \alpha$.

In summary, the maximum entropy case with known average number of Byzantines, corresponds to assuming i.i.d. node states for which the probability of being malicious is constant and known to the FC[2]. It can also be noticed that when $\alpha = 0.5$, we go back to the unconstrained maximum entropy case discussed in the previous section.

Let us assume, then, that A_i's are Bernoulli random variables with parameter α, i.e., $P_{A_i}(1) = \alpha$, $\forall i$. In this way, the number of Byzantines in the network is a random variable with a binomial distribution. In particular, we have $P(a^n) = \prod_i P(a_i)$, and hence (5.4) can be rewritten as:

$$s^{m,*} = \arg\max_{s^m} \sum_{a^n} \left(\prod_{i=1}^{n} P(\mathbf{r}_i | a_i, s^m) P(a_i) \right). \tag{5.11}$$

The expression in round brackets corresponds to a factorization of $P(\mathbf{r}, a^n|s^m)$. If we look at that expression as a function of a^n, it is a product of marginal functions. By exploiting the distributivity of the product with respect to the sum we can rewrite (5.11) as follows

$$s^{m,*} = \arg\max_{s^m} \prod_{i=1}^{n} \left(\sum_{a_i \in \{0,1\}} P(\mathbf{r}_i | a_i, s^m) P(a_i) \right), \tag{5.12}$$

which can be computed more efficiently, especially for large n. The expression in (5.12) can also be derived directly from (5.2) by exploiting first the independence of the reports and then applying the law of total probability. By reasoning as we did to derive (5.8), the to-be-maximized expression for

[1] For any $p \leq 1$ we have: $h(p) = p \log_2 p + (1-p) \log_2 (1-p)$.

[2] Sometimes this scenario is referred to as Clairvoyant case [112].

the case of symmetric error probabilities at the nodes becomes

$$s^{m,*} = \arg\max_{s^m} \prod_{i=1}^{n} \left[(1-\alpha)(1-\varepsilon)^{m_{eq}(i)} \varepsilon^{m-m_{eq}(i)} + \alpha(1-\delta)^{m_{eq}(i)} \delta^{m-m_{eq}(i)} \right]. \tag{5.13}$$

Due to the independence of node states, the complexity of the above maximization problem grows only linearly with n, while it is exponential with respect to m, since it requires the evaluation of the to-be-minimized function for all possible sequence s^m. For this reason, the optimal fusion strategy can be adopted only when the length of the observed sequence is limited.

5.2.2.2 Maximum entropy with $N_B < h$

As a second possibility, the FC is assumed to know only that the number of Byzantines N_B is lower than a certain value h ($h \leq n$). For instance, as already observed in previous works [33, 110, 112], when the number of Byzantines exceeds the number of honest nodes no meaningful decision can be made. Then, as a worst case assumption, it makes sense for the FC to assume that $N_B < n/2$ (i.e., $h = n/2$), since if this is not the case, no correct decision can be made anyhow. Under this assumption, the maximum entropy distribution is the one which assigns exactly the same probability to all the sequences a^n for which $\sum_i a_i < n/2$. More in general, the FC might have some a priori knowledge on the maximum number of corrupted (or corruptible) links in the network, and then he can constraint N_B to be lower than h with $h < n/2$. To derive the optimum fusion strategy in this setting, let $\mathcal{I}$ be the indexing set $\{1, 2, ..., n\}$. Let $\mathcal{I}_k$ be the set of all the possible k-subsets of $\mathcal{I}$. Let $I \in \mathcal{I}_k$ be a random variable with the indexes of the byzantine nodes, a node i being Byzantine if $i \in I$, honest otherwise. With this notation, we can rewrite (5.3) as

$$s^{m,*} = \arg\max_{s^m} \sum_{k=0}^{h-1} \sum_{I \in \mathcal{I}_k} P(\mathbf{r}|I, s^m) P(s^m), \tag{5.14}$$

where the term $P(I)$ (or equivalently $P(a^n)$) is omitted since all the sequences for which $N_B < h$ have the same probability. In the case of symmetric

local error probabilities, (5.14) takes the following form:

$$s^{m,*} = \arg\max_{s^m} \sum_{k=0}^{h-1} \sum_{I \in \mathcal{I}_k} \left(\prod_{i \in I} (1-\delta)^{m_{eq}(i)} \delta^{m-m_{eq}(i)} \prod_{i \in \mathcal{I} \setminus I} (1-\varepsilon)^{m_{eq}(i)} \varepsilon^{m-m_{eq}(i)} \right). \tag{5.15}$$

Since, reasonably, h is a fraction of n, a problem with (5.15) is the complexity of the inner summation, which grows exponentially with n (especially for values of k close to h). Together with the maximization over all possible s^m, this results in a doubly exponential complexity, making the direct implementation of (5.15) problematic. In Section 5.3, we present an efficient algorithm based on dynamic programming which reduces the computational complexity of the maximization in (5.15).

We conclude by stressing an important difference between the case considered in this subsection and the maximum entropy case with fixed $E[N_B]$, with the same average number of Byzantines. In the setting with a fixed $E[N_B]$ ($< n/2$) there is no guarantee that the number of Byzantines is always lower than the number of honest nodes, as it is the case in the setting analyzed in this subsection when $h \leq n/2$. This observation will be crucial to explain some of the results that we will present later on in the chapter.

5.2.3 Fixed number of Byzantines

The final setting we are going to present assumes that the fusion center knows the exact number of Byzantines, say n_B. This is a more favorable situation with respect to those addressed so far. The derivation of the optimum decision fusion rule stems from the observation that, in this case, $P(a^n) \neq 0$ only for the sequence for which $\sum_i a_i = n_B$. For such sequences, $P(a^n)$ is constant and equal to $\binom{n}{n_B}^{-1}$. By using the same notation used in the previous section, the optimum fusion rules, then, is:

$$s^{m,*} = \arg\max_{s^m} \sum_{I \in \mathcal{I}_{n_B}} P(\mathbf{r}|I, s^m) p(s^m), \tag{5.16}$$

which reduces to

$$s^{m,*} = \arg\max_{s^m} \sum_{I \in \mathcal{I}_{n_B}} \left(\prod_{i \in I} (1-\delta)^{m_{eq}(i)} \delta^{m-m_{eq}(i)} \prod_{i \in \mathcal{I} \setminus I} (1-\varepsilon)^{m_{eq}(i)} \varepsilon^{m-m_{eq}(i)} \right), \tag{5.17}$$

in the case of equal local error probabilities. With regard to computational complexity, even if the summation over all possible number of Byzantines is no more present, the direct implementation of (5.17) is still very complex due to the exponential dependence of the cardinality of $\mathcal{I}_{n_B}$ with respect to n.

5.3 An efficient implementation based on dynamic programming

The computational complexity of a direct implementation of (5.15) and (5.17) hinders the derivation of the optimum decision fusion rule for large size networks. Specifically, the problem with (5.15) and (5.17) is the exponential number of terms of the summation over $\mathcal{I}_k$ ($\mathcal{I}_{n_B}$ in (5.17)). In this section, we show that an efficient implementation of such summations is possible based on Dynamic Programming (DP) [118].

Dynamic programming is an optimization strategy which allows to solve complex problems by transforming them into subproblems and by taking advantage of the subproblems overlap in order to reduce the number of operations. When facing with complex recursive problems, by using dynamic programming we solve each different subproblem only once by storing the solution for subsequent use. If during the recursion the same subproblem is encountered again, the problem is not solved twice since its solution is already available. Such a re-use of previously solved subproblems is often referred in literature as memoization algorithm [118]. Intuitively, DP allows to reduce the complexity of problems with a structure, such that the solutions of the same subproblems can be reused many times.

We now present how dynamic programming can be used to reduce the complexity of the problem in this chapter. Let us focus on a fixed k (and n)

and let us define the function $f_{n,k}$ as follows:

$$f_{n,k} = \sum_{I \in \mathcal{I}_k} \left(\prod_{i \in I} (1-\delta)^{m_{eq}(i)} \delta^{m-m_{eq}(i)} \prod_{i \in \mathcal{I} \setminus I} (1-\varepsilon)^{m_{eq}(i)} \varepsilon^{m-m_{eq}(i)} \right). \quad (5.18)$$

By focusing on node i, there are some configurations $I \in \mathcal{I}_k$ for which such a node belongs to I, while for others the node belongs to the complementary set $\mathcal{I} \setminus I$. Let us define $b(i) = (1-\delta)^{m_{eq}(i)} \delta^{m-m_{eq}(i)}$ and $h(i) = (1-\varepsilon)^{m_{eq}(i)} \varepsilon^{m-m_{eq}(i)}$, which are the two contributions that node i can provide to each term of the sum, depending on whether it belongs to $\mathcal{I}$ or $\mathcal{I} \setminus I$. Let us focus on the first indexed node. By exploiting the distributivity of the product with respect to the sum, expression (5.18) can be rewritten in a recursive manner as:

$$f_{n,k} = b(1) f_{n-1,k-1} + h(1) f_{n-1,k}. \quad (5.19)$$

By focusing on the second node, we can iterate on $f_{n-1,k-1}$ and $f_{n-1,k}$, getting:

$$f_{n-1,k-1} = b(2) f_{n-2,k-2} + h(2) f_{n-2,k-1}, \quad (5.20)$$

and

$$f_{n-1,k} = b(2) f_{n-2,k-1} + h(2) f_{n-2,k}. \quad (5.21)$$

it can be noticed that subfunction $f_{n-2,k-1}$ appears in both (5.20) and (5.21) and then it can be computed only once. The procedure can be iterated for each subfunction until we reach a subfunction whose value can be computed in closed form, that is: $f_{r,r} = \prod_{i=n-r+1}^{n} b(i)$ and $f_{r,0} = \prod_{i=n-r+1}^{n} h(i)$, for some $r \leq k$. By applying the memorization strategy typical of dynamic programming, the number of required computations is given by the number of nodes in the tree depicted in Figure 5.2, where the leaves correspond to the terms computable in closed form[3]. By observing that the number of the nodes of the tree is $k(k+1)/2 + k(n-k-k) + k(k+1)/2 = k(n-k+1)$, we conclude that the number of operations is reduced from $\binom{n}{k}$ to $k(n-k+1)$, which corresponds to a quadratic complexity instead of an exponential one.

[3]The figure refers to the case $k < n-k$, which is always the case in our setup since $k < \lfloor n/2 \rfloor$.

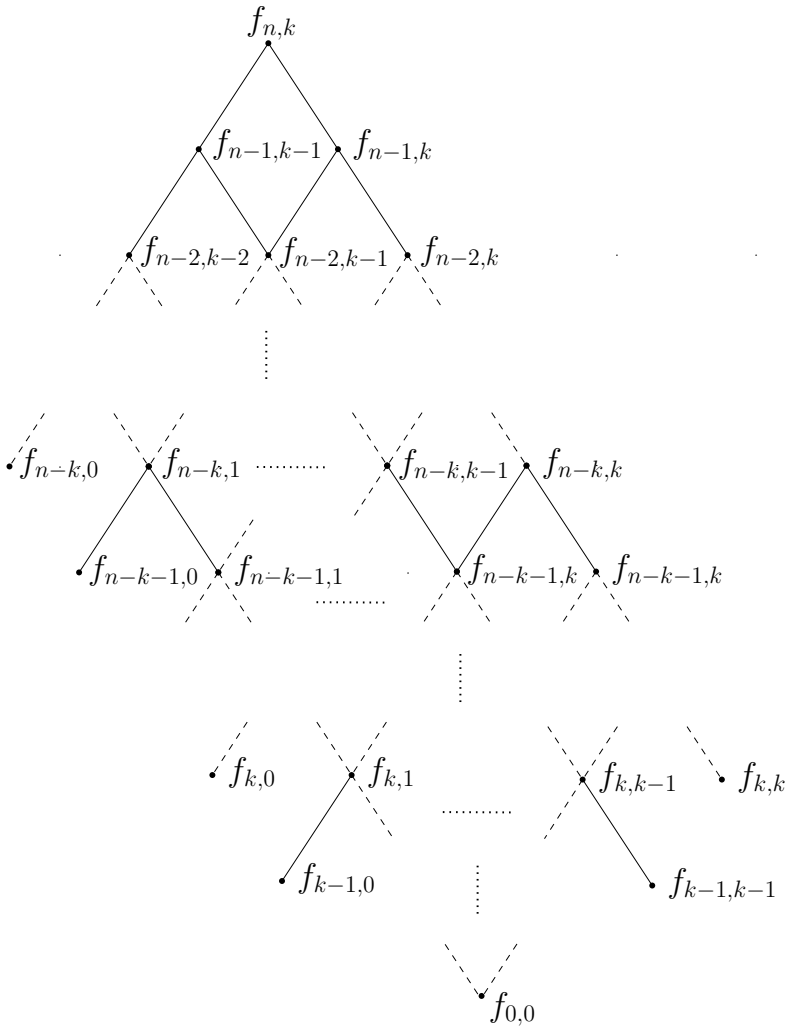


Figure 5.2: *Efficient implementation of the function in* (5.18) *based on dynamic programming. The figure depicts the tree with the iterations for the case* $k < n - k$.

5.4 Decision fusion with Byzantines game

BT: I would rephrase with 'Decision Fusion Game' (or 'Optimum Decision Fusion Game'). Unless we go for 'Optimum Decision Fusion Game', the title

of this section turns out to be the same or very similar to the title used for the previous game (Section 4.4.1 and 4.4.2). Even if they are different chapters, we may prefer to differentiate (for instance, for the previous game we could use the title 'The Decision Fusion Game with Isolation of Byzantines'....)

The optimum decision fusion rules derived in Section 5.2 assume that the FC knows the attack strategy adopted by the Byzantines, which in the simplified case studied in this chapter corresponds to knowing P_{mal}. By knowing P_{mal}, in fact, the FC can calculate the value of δ used in Equations (5.8), (5.13), (5.15) and (5.17), and hence implement the optimum fusion rule. In previous works, as in [112, 114], it is often conjectured that $P_{\text{mal}} = 1$. In some particular settings, as the ones addressed in [117] and Chapter 4, it has been shown that this choice permits to the Byzantines to maximize the error probability at the fusion center. Such an argument, however, does not necessarily hold when the fusion center can localize the byzantine nodes with good accuracy and when it knows that the byzantine nodes always flip the local decision. In such a case, in fact, the FC can revert the action of the Byzantines by simply inverting the reports received from such nodes, as it is implicitly done by the optimal fusion rules derived in the previous section. In such a situation, it is easy to argue that it is better for the Byzantines to let $P_{\text{mal}} = 0.5$ since in this way the mutual information between the system status and the reports received from the byzantine nodes is equal to zero. In general, the byzantine nodes must face the following dilemma: is it better to try to force the FC to make a wrong decision by letting $P_{\text{mal}} = 1$ and run the risk that if their location in the network is detected the FC receives some useful information from the corrupted reports, or erase the information that the FC receives from the attacked nodes by reducing to zero the mutual information between the corrupted reports and S^m ?

Given the above discussion, it is clear that the FC cannot assume that the Byzantines use $P_{\text{mal}} = 1$, hence making the actual implementation of the optimum decision fusion rule impossible.

In order to exit this apparent deadlock, the race of arms between the Byzantines and the FC is modeled as a two-player, zero-sum, strategic game, whose equilibrium identifies the optimum choices for the FC and the Byzantines. In this model, the interplay is between the value of P_{mal} adopted by

the Byzantines and the value used by the FC in its attempt to implement the optimum fusion rule as game. For sake of clarity, in the following the flipping probability adopted by the Byzantines is indicated as P^B_{mal}, while the symbol P^{FC}_{mal} is used to indicate the value adopted by the FC in its implementation of the optimum fusion rule. With the above ideas in mind, we present the following Decision Fusion Game.

Definition 2. *The $DF_{Byz}(\mathcal{S}_B, \mathcal{S}_{FC}, v)$ game is a two player, zero-sum, strategic, game played by the FC and the Byzantines (collectively acting as a single player), defined by the following strategies and payoff.*

- *The sets of strategies the Byzantines and the FC can choose from are, respectively, the set of possible values of P^B_{mal} and P^{FC}_{mal}:*

$$\begin{aligned} \mathcal{S}_B &= \{P^B_{\text{mal}} \in [0,1]\}; \\ \mathcal{S}_{FC} &= \{P^{FC}_{\text{mal}} \in [0,1]\}. \end{aligned} \tag{5.22}$$

- *The payoff function is defined as the error probability at the FC, indicated as P_e*

$$v = P_e = P(s^* \neq s). \tag{5.23}$$

  where s is the true system state and s^* is the decision made by FC. Of course the Byzantines aim at maximizing P_e, while the FC aims at minimizing it.

Note that according to the definition of DF_{Byz}, the sets of strategies available to the FC and the Byzantines are continuous sets. In practice, however, as done in Chapter 4, the continuous variables P^B_{mal} and P^{FC}_{mal} can be replaced by discrete variables by properly quantizing them, thus getting a game with finite set of strategies for the players that can be represented in normal form.

In the next section, numerical simulations are used to derive the equilibrium point of various versions of the game obtained by varying the probability distribution of Byzantines as detailed in Section 5.2. As we will see, while some versions of the game has a unique Nash (or even dominant) equilibrium in pure strategies, in other cases, a Nash equilibrium exists only in mixed strategies.

5.5 Simulation results and discussion

In order to investigate the behavior of the DF_{Byz} game for different setups and analyze the achievable performance when the FC adopts the optimum decision strategy with parameters tuned following a game-theoretic approach, extensive numerical simulations are used. The first goal of the simulations was to study the existence of an equilibrium point in pure or mixed strategies, and analyze the expected behavior of the FC and the Byzantines at the equilibrium. The second goal was to evaluate the payoff at the equilibrium as a measure of the best achievable performance of Decision Fusion in the presence of Byzantines. Such a value is used to compare the performance of the game-theoretic approach with respect to other approaches.

5.5.1 Analysis of the equilibrium point of the DF_{Byz} game

As anticipated, the first goal of the simulations was to determine the existence of an equilibrium point for the DF_{Byz} game with quantized sets of strategies for FC and Byzantines, namely $\mathcal{S}_{FC}^{q}$ and $\mathcal{S}_{B}^{q}$. The set of available strategies that we considered after the quantization are: $\mathcal{S}_{B}^{q} = \{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and $\mathcal{S}_{FC}^{q} = \{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. The analysis is restricted to values larger than or equal to 0.5 since it is easily arguable that such values always lead to better performance for the Byzantines[4]. As to the choice of the quantization step, it is set to 0.1 to ease the description of the results.

Let $\mathbf{V}$ denote the payoff matrix of the game, that is, the matrix of the error probabilities for each pair of strategies $(P_{\text{mal}}^{B}, P_{\text{mal}}^{FC}) \in \mathcal{S}_{FC}^{q} \times \mathcal{S}_{B}^{q}$. For each setting, the payoff matrix $\mathbf{v}$ is obtained by running the simulations for all the possible moves of FC and the Byzantines.

Sometimes (when the game can be solved with pure strategies), the equilibrium point easily comes out through inspection of the payoff matrix, especially when a rationalizable equilibrium exists. In the general case, we can find equilibrium point by relying on the Minimax Theorem [67]. Let p_B (res. p_{FC}) be a column vector with the probability distribution over the possible values of P_{mal}^{B} (res. P_{mal}^{FC}). The mixed strategies Nash equilibrium (p_B^*, p_{FC}^*)

[4]By using a game-theoretic terminology, this is equivalent to say that the strategies corresponding to $P_{\text{mal}}^{B} < 0.5$ are dominated strategies and hence can be eliminated.

can be found by solving separately the max-min and min-max problems:

$$p_B^* = \arg\max_{p_B(\mathcal{S}_B^q)} \min_{p_{FC}(\mathcal{S}_{FC}^q)} p_B^T \mathbf{v} p_{FC}$$
$$p_{FC}^* = \arg\min_{p_{FC}(\mathcal{S}_{FC}^q)} \max_{p_B(\mathcal{S}_B^q)} p_B^T \mathbf{v} p_{FC} \tag{5.24}$$

which, as we said in Chapter 2, can be reduced to the solution of a Linear Programming problem.

Among all the parameters of the game, the value of m has a major impact on the equilibrium point. The value of m, in fact, determines the ease with which the FC can localize the byzantine nodes, and hence plays a major role in determining the optimum attacking strategy. For this reason, we split our analysis in two parts: the former refers to small values of m, the latter to intermediate values of m. Unfortunately, the exponential growth of the complexity of the optimum decision fusion rule as a function of m prevented us from running simulations with large values of m.

Simulations were carried out by adopting the following setup. The nuumber of trials used to compute P_e at each row of the matrix is 50,000. In particular, for each P_{mal}^B, we used the same 50,000 states to compute P_e for all P_{mal}^{FC} strategies. In all the simulations, $P_{S_j}(0) = P_{S_j}(1) = 0.5$ is used, $n = 20$, and $\varepsilon = 0.1$. The linear programming tools from Matlab Optimization Toolbox [119] is used to solve (5.24).

5.5.1.1 Small m BT: I would prefer a more self-explanatory title, e.g. 'Small observation window/Short observation sequence'

For the first set of simulations, we present a rather low value of m, namely $m = 4$. The other parameters of the game were set as follows: $n = 20$, $\varepsilon = 0.1$. With regard to the number of Byzantines present in the network $\alpha = \{0.3, 0.4, 0.45\}$ is used for the case of independent node states studied in Section 5.2.2.1, and $n_B = \{6, 8, 9\}$ for the case of known number of Byzantines (Section 5.2.3). Such values were chosen so that in both cases we have the same average number of Byzantines, thus easing the comparing between the two settings.

Tables 5.1 through 5.3 report the payoff for all the profiles resulting from the quantized values of P^B_{mal} and P^{FC}_{mal}, for the case of independent node states (constrained maximum entropy distribution). The error probabilities in all the tables are scaled by a convenient power of 10. In all the cases $P^B_{\text{mal}} = 1$ is a dominant strategy for the Byzantines, and the profile $(1, 1)$ is the unique rationalizable equilibrium of the game. As expected, the error probability increases with the number of Byzantines. The value of the payoff at the equilibrium ranges from $P_e = 0.0349$ with $\alpha = 0.3$ to $P_e = 0.3314$ with $\alpha = 0.45$. For completeness, we report the value of the error probability in the non-adversarial setup, which is $P_e = 0.34 \cdot 10^{-5}$. Concerning the computation of the equilibrium point, we can observe that by looking at these tables, apparently the strategy $P^B_{\text{mal}} = 1$ is dominant for the Byzantines, so it is sufficient to consider only the row corresponding to this strategy from the payoff matrix. In this way, the problem becomes monodimensional one and the equilibrium point is the point that minimizes the P_e at the FC, which is $P^{FC}_{\text{mal}} = 1$. This strategy for computing the equilibrium point can be applied whenever we have a dominant strategy for any of the players, however, in this thesis we used Lemke-Howson algorithm [120] to solve the games.

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.845	0.965	1.1	1.3	1.6	2.1
0.6	1.2	1.1	1.2	1.5e-3	1.8	2.6
0.7	2.2	2.0	1.8	1.8e-3	2.1	3.7
0.8	5.4	5.1	5.0	5.0e-3	5.1	7.7
0.9	16.2	16.1	16.5	16.4	16.0	19.1
1.0	43	43.1	46.9	46.8	41.6	**34.9**

Table 5.1: *Payoff of the DF_{Byz} game ($10^3 \times P_e$) with independent node states with $\alpha = 0.3$, $m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

Tables 5.4 through 5.6 report the payoffs for the case of fixed number of Byzantines, respectively equal to 6, 8 and 9.

When $n_B = 6$, $P^B_{\text{mal}} = 0.5$ is a dominant strategy for the Byzantines, and the profile $(0.5, 0.5)$ is the unique rationalizable equilibrium of the game

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.33	0.37	0.44	0.58	0.73	0.85
0.6	0.60	0.54	0.59	0.70	0.80	1.14
0.7	1.38	1.20	1.19	1.24	1.29	2.40
0.8	3.88	3.56	3.36	3.31	3.35	6.03
0.9	9.93	9.61	9.57	9.55	9.54	11.96
1.0	20.33	20.98	21.70	21.90	21.84	**19.19**

Table 5.2: *Payoff of the DF_{Byz} game ($10^2 \times P_e$) with independent node states with $\alpha = 0.4$, $m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.62	0.69	0.86	1.34	1.70	1.57
0.6	1.23	1.15	1.26	1.84	2.18	2.38
0.7	2.94	2.64	2.57	3.00	3.14	5.33
0.8	7.89	7.39	7.03	6.74	6.81	12.73
0.9	18.45	17.94	17.63	17.08	17.07	22.78
1.0	34.39	34.62	34.84	36.66	36.61	**33.14**

Table 5.3: *Payoff of the DF_{Byz} game ($10^2 \times P_e$) with independent node states with $\alpha = 0.45$, $m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

corresponding to a payoff $P_e = 3.8 \cdot 10^{-4}$. This marks a significant difference with respect to the case of independent nodes, where the optimum strategy for the Byzantines was to let $P^B_{\text{mal}} = 1$. The reason behind the different behavior is that in the case of fixed number of nodes, the a-priori knowledge available at the FC is larger than in the case of independent nodes with the same average number of nodes. This additional information permits to the FC to localize the byzantine nodes, which now cannot use $P^B_{\text{mal}} = 1$, since in this case they would still transmit some useful information to the FC. On the contrary, by letting $P^B_{\text{mal}} = 0.5$ the information received from the byzantine nodes is zero,

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	3.80	**3.80**	4.60	7.60	12.0	29.0
0.6	3.60	3.45	3.90	5.20	8.0	17.0
0.7	3.45	2.80	2.80	3.10	4.40	8.75
0.8	4.10	2.85	2.15	2.05	2.25	3.25
0.9	3.55	2.05	1.40	0.95	0.70	0.75
1.0	2.05	0.90	0.35	0.15	0.05	0.05

Table 5.4: *Payoff of the DF_{Byz} game ($10^4 \times P_e$) with $n_B = 6$, $m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	1.2	1.4	1.9	3.1	6.3	18.9
0.6	1.5	1.4	1.4	2.0	3.7	10.0
0.7	1.4	1.1	0.945	1.1	1.7	4.0
0.8	1.4	0.95	0.715	0.58	0.675	1.2
0.9	2.1	1.4	0.995	0.745	0.71	0.78
1.0	7.3	5.7	5.3	3.7	3.0	2.9

Table 5.5: *Payoff of the DF_{Byz} game ($10^3 \times P_e$) with $n_B = 8$, $m = 4$, $n = 20$, $\varepsilon = 0.1$. No pure strategy equilibrium exists.*

hence making the task of the FC harder. When $n_B = 9$ (Table 5.6), the larger number of Byzantines makes the identification of malicious nodes more difficult and $P^B_{\text{mal}} = 1$ is again a dominant strategy, with the equilibrium of the game obtained at the profile (1,1) with $P_e = 0.0551$. A somewhat intermediate situation is observed when $n_B = 8$ (Table 5.5). In this case, no equilibrium point exists (let alone a dominant strategy) if we consider pure strategies only. On the other hand, when mixed strategies are considered, the game has a unique Nash equilibrium for the strategies reported in Table 5.7 (each row in the table gives the probability vector assigned to the quantized values of P_{mal} by one of the players at the equilibrium). Interestingly the optimum strategy of the Byzantines corresponds to alternate playing $P^B_{\text{mal}} = 1$ and

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.22	0.24	0.33	0.63	1.41	4.13
0.6	0.27	0.24	0.27	0.41	0.78	2.03
0.7	0.32	0.24	0.23	0.26	0.37	0.82
0.8	0.54	0.45	0.39	0.36	0.41	0.59
0.9	2.04	1.87	1.76	1.58	1.56	1.66
1.0	9.48	8.76	8.37	6.72	5.88	**5.51**

Table 5.6: *Payoff of the DF_{Byz} game ($10^2 \times P_e$) with $n_B = 9$, $m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

$P^B_{\text{mal}} = 0.5$, with intermediate probabilities. This confirms the necessity for the Byzantines to find a good trade-off between two alternative strategies: set to zero the information transmitted to the FC or try to push it towards a wrong decision. We also observe that the error probabilities at the equilibrium are always lower than those of the game with independent nodes. This is an expected result, since in the case of fixed nodes the FC has a better knowledge about the distribution of Byzantines.

	0.5	0.6	0.7	0.8	0.9	1.0
$P(P^B_{\text{mal}})$	0.179	0	0	0	0	0.821
$P(P^{FC}_{\text{mal}})$	0	0	0	0.846	0.154	0
$P^*_e = 3.6e-3$						

Table 5.7: *Mixed strategies equilibrium for the DF_{Byz} game with $n_B = 8$, $m = 4$, $n = 20$, $\varepsilon = 0.1$. P^*_e indicates the error probability at the equilibrium.*

The last case we present corresponds to a situation in which the FC knows that the number of Byzantines cannot be larger than a certain value h (see Sec. 5.2.2.2).

We first consider the case in which the FC knows only that the number of Byzantines is lower than $n/2$. The payoff for this instantiation of the DF_{Byz} game is given in Table 5.8. In order to compare the results of this case with

those obtained for the case of independent nodes and that of fixed number of Byzantines, we observe that when all the sequences a^n with $n_B < n/2$ have the same probability, the average number of Byzantines turns out to be 7.86. The most similar settings, then, are that of independent nodes with $\alpha = 0.4$ and that of fixed number of nodes with $n_B = 8$. With respect to the former, the error probability at the equilibrium is significantly smaller, thus confirming the case of independent nodes as the worst scenario for the FC. This is due to the fact that with $\alpha = 0.4$ it is rather likely that number of Byzantines is larger than 0.5 thus making any reliable decision impossible. The error probability obtained with a fixed number of Byzantines equal to 8, however, is much lower. This is a reasonable result, since in that case the a-priori information available to the FC permits better localization of the corrupted reports.

We now move to the case with $h < n/2$. Table 5.9 reports the payoffs of the game when $N_B < n/3$. By assuming a maximum entropy distribution over the admissible configurations a^n with $N_B < n/3$, the average number of Byzantines turns out to be 4.64. In this case, the equilibrium point shifts to $(0.5, 0.5)$. This confirms the behavior discussed in the previous paragraph: since the average number of Byzantines is lower the FC is able to localize them with a better accuracy, then it is better for the Byzantines to minimize the information delivered to the FC.

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.15	0.17	0.20	0.29	0.39	0.51
0.6	0.17	0.16	0.16	0.22	0.29	0.40
0.7	0.19	0.15	0.14	0.16	0.20	0.30
0.8	0.27	0.20	0.17	0.16	0.17	0.22
0.9	0.85	0.76	0.72	0.63	0.58	0.63
1.0	3.81	3.49	3.30	2.62	2.24	**2.13**

Table 5.8: *Payoff of the DF_{Byz} game ($10^2 \times P_e$) with $N_B < n/2$. The other parameters of the game are set as follows: $m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	**1.9**	2.10	2.30	2.85	3.4	4.05
0.6	1.85	1.75	1.9	2.0	2.85	3.80
0.7	1.3	1.05	0.75	0.8	1.30	2.20
0.8	1.7	1.45	1.15	1.1	1.15	1.50
0.9	1.25	0.65	0.5	0.35	0.35	0.35
1.0	0.85	0.6	0.4	0.1	0.05	0.05

Table 5.9: *Payoff of the DF_{Byz} game ($10^4 \times P_e$) with $N_B < n/3$. The other parameters of the game are set as follows: $m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

5.5.1.2 Intermediate values of m BT: Observation sequences/windows of intermediate length

In this section we report the results that we got when the length of the observation vector increases. It is expected that by comparing the reports sent by the nodes corresponding to different components of the state vector allows a better identification of the byzantine nodes, thus modifying the equilibrium of the game. Specifically, the simulations carried out in the previous section are repeated, by letting $m = 10$. Though desirable, repeating the simulations with even larger values of m is not possible due to the exponential growth of the complexity of the optimum fusion rule with m.

Tables 5.10 through 5.12 report the payoffs of the game for the case of independent node states. As it can be seen, $P^B_{\text{mal}} = 1.0$ is still a dominant strategy for the Byzantines and the profile (1,1) is the unique rationalizable equilibrium of the game. Moreover, the value of P_e at the equilibrium is slightly lower than for $m = 4$, when $\alpha = 0.3$ and $\alpha = 0.4$ (see Tables 5.1 and 5.2). Such an advantage disappears when $\alpha = 0.45$ (see Table 5.3), since the number of Byzantines is so large that identifying them is difficult even with $m = 10$.

The results of the simulations for the case of fixed number of nodes with $n_B = \{6, 8, 9\}$ are given in Tables 5.13 through 5.15. With respect to the case of $m = 4$, the optimum strategy for the Byzantines shifts to $P^B_{\text{mal}} = 0.5$.

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.258	0.28	0.39	0.63	1.0	1.7
0.6	0.28	0.226	0.248	0.362	0.652	2.0
0.7	0.346	0.22	0.206	0.23	0.314	5.3
0.8	1.2	0.648	0.44	0.428	0.498	13.9
0.9	8.6	7.8	7.6	7.8	7.5	19.9
1.0	41.9	46.7	50.9	59.8	52.2	**32.9**

Table 5.10: *Payoff of the DF_{Byz} game ($10^3 \times P_e$) with independent node states with $\alpha = 0.3$, $m = 10$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.11	0.13	0.19	0.73	2.16	0.68
0.6	0.11	8.32e-2	9.96e-2	0.26	0.67	1.30
0.7	0.18	7.66e-2	6.62e-2	9.52e-2	0.18	4.87
0.8	1.10	0.60	0.33	0.24	0.28	10.41
0.9	5.77	4.75	3.95	3.53	3.41	13.44
1.0	20.41	21.26	22.65	24.27	26.21	**18.72**

Table 5.11: *Payoff of the DF_{Byz} game ($10^2 \times P_e$) with independent node states with $m = 10$, $n = 20$, $\alpha = 0.4$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

When $n_B = 6$, $P^B_{\text{mal}} = 0.5$ is a dominant strategy, while for $n_B = 8$ and $n_B = 9$, no equilibrium point exists if we consider only pure strategies. The mixed strategy equilibrium point for these cases is given in Tables 5.18 and 5.19. By comparing those tables with those of the case $m = 4$, the preference towards $P^B_{\text{mal}} = 0.5$ is evident.

Table 5.16, gives the results for the case $N_B < n/2$. As in the case of fixed number of Byzantines, the equilibrium point strategy passes from the pure strategy (1,1) to a mixed strategy (see Table 5.20). Once again, the reason for such a behavior, is that when m increases, the amount of information available

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.20	0.23	0.47	2.88	10.92	1.26
0.6	0.22	0.18	0.24	0.80	2.85	2.93
0.7	0.50	0.19	0.15	0.23	0.65	10.64
0.8	2.61	1.24	0.63	0.41	0.59	20.65
0.9	11.74	9.28	7.08	5.65	5.21	25.85
1.0	34.25	34.94	36.01	37.74	39.87	**33.17**

Table 5.12: *Payoff of the DF_{Byz} game ($10^2 \times P_e$) with independent node states with $\alpha = 0.45$, $m = 10$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	1.22	**1.22**	1.40	2.20	5.06	11.0
0.6	1.12	0.94	1.02	1.26	2.56	5.34
0.7	1.22	0.58	0.56	0.64	0.98	2.06
0.8	1.22	0.36	0.32	0.28	0.30	0.56
0.9	1.40	0.20	0.18	0.16	0.10	0.18
1.0	1.52	0.14	0.14	0.10	6e-2	4e-2

Table 5.13: *Payoff of the DF_{Byz} game ($10^4 \times P_e$) with $n_B = 6$, $m = 10$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

to the FC increases, hence making the detection of corrupted reports easier. As a result, the Byzantines must find a trade-off between forcing a wrong decision and reducing the mutual information between the corrupted reports and system states. Eventually, Table 5.17 reports the results of the game for the case $N_B < n/3$ and $m = 10$. As one could expect, the profile $(0.5, 0.5)$ is still the equilibrium point of the game, as the optimum strategy for the Byzantines continues to be the one which minimizes the amount of information delivered to the FC. We conclude observing that even with $m = 10$, the case of independent nodes results in the worst performance.

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	4.04	4.44	6.24	10.0	24.0	71.0
0.6	4.02	3.30	3.58	5.24	10.0	26.0
0.7	3.48	2.16	2.14	2.16	3.26	7.76
0.8	3.56	1.10	0.88	0.78	0.98	2.08
0.9	4.60	0.68	0.54	0.30	0.26	0.44
1.0	5.20	0.54	0.20	8e-2	0	0

Table 5.14: *Payoff of the DF_{Byz} game ($10^4 \times P_e$) with $n_B = 8$, $m = 10$, $n = 20$, $\varepsilon = 0.1$. No pure strategy equilibrium exists.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	6.74	7.82	12	23	52	168
0.6	5.44	4.94	6.14	9.40	18	52
0.7	4.22	3.30	2.78	3.38	5.86	15
0.8	3.0	2.24	1.24	0.78	1.32	3.24
0.9	5.22	2.36	1.34	1.02	0.88	1.24
1.0	70	40	19	8.90	3.44	2.42

Table 5.15: *Payoff of the DF_{Byz} game ($10^4 \times P_e$) with $n_B = 9$, $m = 10$, $n = 20$, $\varepsilon = 0.1$. No pure strategy equilibrium exists.*

5.5.2 Performance at the equilibrium

As a last analysis we present a comparison between the error probability obtained by the game-theoretic optimum decision fusion presented in this chapter with other different approaches presented in the previous chapter. Specifically, the scheme of this chapter is compared against a simple majority-based decision fusion rule according to which the FC decides that $s_j = 1$ if and only if $\sum_i r_{ij} > n/2$ (Maj), against the hard isolation and the soft isolation schemes.

In order to carry out a fair comparison and to take into account the game-theoretic nature of the problem, the performance of all the schemes are evaluated at the equilibrium. For the HardIS and SoftIS schemes this corresponds

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	4.46	5.38	6.64	9.88	16	27
0.6	3.90	3.38	4.10	5.90	9.42	19
0.7	3.04	2.24	1.82	2.26	3.68	7.28
0.8	2.78	1.72	1.0	0.72	0.90	1.70
0.9	3.24	1.38	0.62	0.30	0.20	0.48
1.0	27	15	6.84	4.68	1.42	1.04

Table 5.16: *Payoff of the DF_{Byz} game ($10^4 \times P_e$) with $N_B < n/2$. The other parameters of the game are set as follows: $m = 10$, $n = 20$, $\varepsilon = 0.1$. No pure strategy equilibrium exists.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	**0.5**	0.58	0.66	0.78	1.1	1.56
0.6	0.44	0.42	0.48	0.56	0.88	1.3
0.7	0.48	0.48	0.46	0.48	0.54	0.86
0.8	0.4	0.36	0.3	0.22	0.26	0.26
0.9	0.34	0.3	0.22	0.16	0.012	0.016
1.0	0.34	0.28	0.16	0.06	0.02	0.02

Table 5.17: *Payoff of the DF_{Byz} game ($10^4 \times P_e$) with $N_B < n/3$ in the following setup: $m = 10$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

	0.5	0.6	0.7	0.8	0.9	1.0
$P(P^B_{\text{mal}})$	0.921	0	0	0	0	0.079
$P(P^{FC}_{\text{mal}})$	0.771	0.229	0	0	0	0
$P^*_e = 4.13e-4$						

Table 5.18: *Mixed strategies equilibrium for the DF_{Byz} game with $n_B = 8$, $m = 10$, $n = 20$, $\varepsilon = 0.1$. P^*_e indicates the error probability at the equilibrium.*

	0.5	0.6	0.7	0.8	0.9	1.0
$P(P^B_{\text{mal}})$	0.4995	0	0	0	0	0.5005
$P(P^{FC}_{\text{mal}})$	0	0	0.66	0.34	0	0
$P^*_e = 1.58e-3$						

Table 5.19: *Mixed strategies equilibrium for the DF_{Byz} game with $n_B = 9$, $m = 10$, $n = 20$, $\varepsilon = 0.1$. P^*_e indicates the error probability at the equilibrium.*

	0.5	0.6	0.7	0.8	0.9	1.0
$P(P^B_{\text{mal}})$	0.4	0	0	0	0	0.6
$P(P^{FC}_{\text{mal}})$	0	0	0.96	0.04	0	0
$P^*_e = 6.76e-4$						

Table 5.20: *Mixed strategies equilibrium for the DF_{Byz} game with $N_B < n/2$ with $m = 10$, $n = 20$, $\varepsilon = 0.1$. P^*_e indicates the error probability at the equilibrium.*

to letting $P^B_{\text{mal}} = 1$. In fact, in the previous chapter, it is shown that this is a dominant strategy for these two specific fusion schemes. As a consequence, P^{FC}_{mal} is also set to 1, since the FC knows in advance that the Byzantines will play the dominant strategy. For the Maj fusion strategy, the FC has no degrees of freedom, so no game actually exists in this case. With regard to the Byzantines, it is easy to realize that the best strategy is to let $P^B_{\text{mal}} = 1$. When the equilibrium corresponds to a mixed strategy, the error probability is averaged according to the mixed strategies at the equilibrium. Tables 5.21 and 5.22 show the error probability at the equilibrium for the tested systems under different setups. As it can be seen, the fusion scheme resulting for the application of the optimum fusion rule in a game-theoretic setting, consistently provides better results for all the analyzed cases. Expectedly, the improvement is more significant for the setups in which the FC has more information about the distribution of the Byzantines across the network.

	Maj	HardIS	SoftIS	OPT
Independent nodes, $\alpha = 0.3$	0.073	0.048	0.041	0.035
Independent nodes, $\alpha = 0.4$	0.239	0.211	0.201	0.192
Independent nodes, $\alpha = 0.45$	0.362	0.344	0.338	0.331
Fixed n. of nodes $n_B = 6$	0.017	0.002	6.2e-4	3.8e-4
Fixed n. of nodes $n_B = 8$	0.125	0.044	0.016	0.004
Fixed n. of nodes $n_B = 9$	0.279	0.186	0.125	0.055
Max entropy with $N_B < n/2$	0.154	0.086	0.052	0.021
Max entropy with $N_B < n/3$	0.0041	5e-4	2.15e-4	1.9e-4

Table 5.21: *Error probability at the equilibrium for various fusion schemes. All the results have been obtained by letting* $m = 4$, $n = 20$, $\varepsilon = 0.1$.

	Maj	HardIS	SoftIS	OPT
Independent nodes, $\alpha = 0.3$	0.073	0.0364	0.0346	0.033
Independent nodes, $\alpha = 0.4$	0.239	0.193	0.19	0.187
Independent nodes, $\alpha = 0.45$	0.363	0.334	0.333	0.331
Fixed n. of nodes $n_B = 6$	0.016	1.53e-4	1.41e-4	1.22e-4
Fixed n. of nodes $n_B = 8$	0.126	0.0028	9.68e-4	4.13e-4
Fixed n. of nodes $n_B = 9$	0.279	0.0703	0.0372	1.58e-3
Max entropy with $N_B < n/2$	0.154	0.0271	0.0141	6.8e-4
Max entropy with $N_B < n/3$	0.0039	9.8e-05	7.40e-05	5e-05

Table 5.22: *Error probability at the equilibrium for various fusion schemes. All the results have been obtained by letting* $m = 10$, $n = 20$, $\varepsilon = 0.1$.

5.5.3 Assumptions validation and discussion

In many real life situations, the information about the Byzantines distribution in the network is not available or easy to be obtained. For this purpose, in this subsection, we discuss the hypothesis about the FC's knowledge of the Byzantines distribution when applying the optimum fusion rule. The performance of the optimum rule is evaluated and the performance loss assessed

under a mismatch between the Byzantines distribution considered by the FC and the true distribution.

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	9.35e-4	8.75e-4	0.001	0.0012	0.0015	0.0019
0.6	0.0011	9.85e-4	0.001	0.0012	0.0015	0.0019
0.7	0.0021	0.0019	0.0018	0.0018	0.0022	0.0032
0.8	0.0057	0.0053	0.0052	0.0052	0.0059	0.0077
0.9	0.0160	0.0158	0.0157	0.0157	0.0163	0.0187
1.0	0.0414	0.0413	0.0413	0.0413	0.0372	**0.0350**

Table 5.23: *Payoff of the DF_{Byz} game with independent node states with $\alpha_{FC} = 0.2$, $\alpha = 0.3$,$m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.0036	0.0035	0.0041	0.0047	0.0057	0.0071
0.6	0.0067	0.0061	0.0061	0.0064	0.0080	0.0104
0.7	0.0153	0.0143	0.0139	0.0139	0.0173	0.0242
0.8	0.0396	0.0382	0.0379	0.0379	0.0448	0.0578
0.9	0.1013	0.1005	0.1003	0.1002	0.1081	0.1201
1.0	0.2040	0.2039	0.2039	0.2039	0.1965	**0.1927**

Table 5.24: *Payoff of the DF_{Byz} game with independent node states with $\alpha_{FC} = 0.2$, $\alpha = 0.4$,$m = 4$, $n = 20$, $\varepsilon = 0.1$. The equilibrium point is highlighted in bold.*

In Table 5.23, the probability that a node is Byzantine is $\alpha = 0.3$ while the estimated value at the FC mismatches with the real value and it is set to $\alpha_{FC} = 0.2$. By comparing this table to Table 5.1, we see that the equilibrium point does not change and the payoff at the equilibrium is 0.0350 for mismatched α and 0.0349 for matched α. Then, in this situation, the loss in performance is very low and the mismatch does not affect heavily the performance of the optimum fusion rule. This loss in performance increases a little

bit when the difference between the α of Byznatines and the one used at the FC increases. For instance, by comparing Table 5.24 to Table 5.2 we can see that the equilibrium point remains at (1,1) but with performance loss is equal 0.0008.

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.0020	0.0019	0.0018	0.0019	0.0023	0.0031
0.6	0.0035	0.0029	0.0025	0.0023	0.0025	0.0030
0.7	0.0057	0.0043	0.0033	0.0029	0.0027	0.0028
0.8	0.010	0.0070	0.0055	0.0047	0.0038	0.0035
0.9	0.0194	0.0134	0.0118	0.0107	0.0091	0.0085
1.0	0.0516	0.0420	0.0410	0.0394	0.0375	**0.0372**

Table 5.25: *Payoff of the DF_{Byz} game with $N_{B_{FC}} < n/4$ in the following setup: $m = 4$, $n = 20$, $\varepsilon = 0.1$, $N_B < n/2$. The equilibrium point is highlighted in bold.*

$P^B_{\text{mal}}/P^{FC}_{\text{mal}}$	0.5	0.6	0.7	0.8	0.9	1.0
0.5	0.0015	0.0015	0.0016	0.0020	0.0027	0.0042
0.6	0.0022	0.0018	0.0017	0.0018	0.0023	0.0034
0.7	0.0029	0.0021	0.0017	0.0016	0.0016	0.0023
0.8	0.0047	0.0033	0.0025	0.0022	0.0019	0.0021
0.9	0.0112	0.0092	0.0081	0.0078	0.0072	0.0069
1.0	0.0401	0.0379	0.0362	0.0342	0.0329	**0.0296**

Table 5.26: *Payoff of the DF_{Byz} game with $N_{B_{FC}} < n/6$ in the following setup: $m = 4$, $n = 20$, $\varepsilon = 0.1$, $N_B < n/2$. The equilibrium point is highlighted in bold.*

Now we consider a different situation that corresponds to maximum entropy with $N_B < h$. In this case, the FC wrongly estimates the maximum number of Byzantines in the network. We consider two cases: in the first case shown in Table 5.25, the maximum number of Byzantines in the network is bounded by $n/4$ while the real Byzantines fraction is bounded by $n/2$ with

an average number of Byzantines of 7.858; In the second case shown in Table 5.26, the same setting was adopted but the maximum number of Byzantines in the network estimated at the FC is bounded by $n/6$. By comparing Table 5.25 and 5.26 with Table 5.8, we see that the equilibrium point is the same but we have a loss in performance of 0.0159 with respect to Table 5.25 and of 0.0083 with respect to Table 5.26.

5.6 Conclusions

We have analyzed the problem of decision fusion in distributed sensor networks in the presence of Byzantines. We first derived the optimum decision strategy by assuming that the statistical behavior of the Byzantines is known. Then we relaxed such an assumption by casting the problem into a game-theoretic framework in which the FC tries to guess the behavior of the Byzantines. The Byzantines, in turn, must fix their corruption strategy without knowing the guess made by the FC. We considered several versions of the game with different distributions of the Byzantines across the network. Specifically, we considered three setups: unconstrained maximum entropy distribution, constrained maximum entropy distribution and fixed number of Byzantines. In order to reduce the computational complexity of the optimum fusion rule for large network sizes, we discussed an efficient implementation based on Dynamic Programming. Simulation results show that increasing the observation window m leads to better identification of the Byzantines at the FC. This forces the Byzantines to look for a trade-off between forcing the FC to make a wrong decision on one hand, and reducing the mutual information between the reports and the system state on the other hand. Simulation results show that, in all the analyzed cases, the performance at the equilibrium are superior to those obtained by previously proposed techniques.

Chapter 6

An Efficient Nearly-Optimum Decision Fusion Technique Based on Message Passing

6.1 Introduction

In the attempt to reduce the computational complexity while minimizing the loss of performance with respect to the optimum fusion rule presented in Chapter 5, in this chapter, we present a near-optimum fusion scheme based on message passing and factor graphs. Moreover, we consider a more general model for the system state that includes both Markovian and independent sequences. The analysis confirms the results of Chapter 5 that the optimum strategy for the Byzantines is to follow a dual-behavior to find a trade-off between inducing global decision error at the FC and avoid being detected by trying to minimize the mutual information between the reports and the sequence of system states.

Specifically, Chapter 5 showed that the complexity of the optimum decision fusion algorithm grows exponentially with the length of the observation window m. Such a complexity prevents the adoption of the optimum decision fusion rule in many practical situations. Also, the results regarding the optimum strategies of the Byzantines and the FC derived in previous chapter can not be immediately applied to the case of large observation windows.

Message passing algorithms, based on the so called Generalised Distributive Law (GDL, [121], [122]), have been widely applied to solve a large range of optimization problems, including decoding of Low Density Parity Check (LDPC) codes [123] and BCJR codes [121], dynamic programming [124], solution of probabilistic inference problems on Bayesian networks [125] (in this case message passing algorithms are known as *belief propagation*). Here, Mes-

sage Passing (MP) is used to introduce a near-optimal solution of the decision fusion problem with multiple observations whose complexity grows only linearly with the size of the observation window, thus marking a dramatic improvement with respect to the exponential complexity of the optimal scheme explained in Chapter 5.

Numerical simulations run by focusing on the case of small observation windows, for which the optimum solution can still be applied, reveal that the MP scheme gives near-optimal performance at a much lower complexity than the optimum scheme. Then, numerical simulations are run to evaluate the performance of the MP method for long observation windows. As a result, we show that, even in this case, the MP scheme maintains the performance improvement over the simple majority rule, the hard isolation and the soft isolation schemes described in Chapter 4.

As opposed to Chapter 5, the analysis is not limited to the case of independent system states, but it is extended to a more realistic scenario where the sequence of states obey a Markovian distribution [126], as depicted in Figure 6.1. The Markovian model is rather common in the case of cognitive radio networks [127–129] where the primary user occupancy of the spectrum is often modelled as a Hidden Markov Model (HMM).

The Markovian case is found to be more favourable for the FC with respect to the case of independent states, due the additional a-priori information available to the FC.

Last but not the least, we confirm that the dual optimum behavior of the Byzantines observed in Chapter 5 is also present in the case of large observation windows, even if in the Markovian case, the Byzantines may continue using the maximum attack power for larger observation windows.

6.2 Notation and Problem Formulation

For the analysis in this chapter the same notation used in Chapter 5 and 4 is adopted. For the sake of clarity, here, we recapitulate such notation. Let $S^m = \{S_1, S_2, \ldots, S_m\}$ with $S_j \in \{0, 1\}$ indicating the sequence of system states over an observation window of length m. The nodes collect information about the system through the vectors $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_n$, with $\mathbf{x}_i$ indicating the

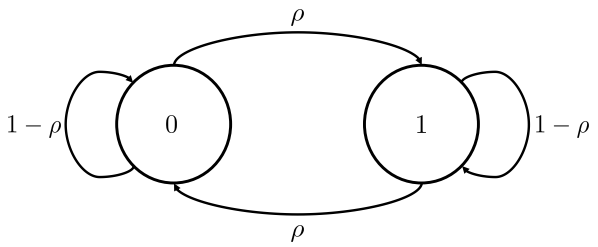


Figure 6.1: *Markovian model for system states. When* $\rho = 0.5$ *subsequent states are independent.*

observations available at node i. Based on such observations, a node i makes a local decision u_{ij} about system state s_j. The local error probability, hereafter indicated as ε, is assumed to be independent of both i and j. The state of the nodes in the network is given by the vector $a^n = \{a_1, a_2, \ldots, a_n\}$ with $a_i = 1/0$ indicating that node i is honest or Byzantine, respectively. Finally, the matrix $\mathbf{R} = \{r_{ij}\}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$ contains all the reports received by the FC. Specifically, r_{ij} is the report sent by node i relative to s_j. For honest nodes we have $u_{ij} = r_{ij}$ while, for Byzantines we have $p(u_{ij} \neq r_{ij}) = P_{\text{mal}}$. The Byzantines corrupt the local decisions independently of each other.

By assuming that the transmission between nodes and fusion center takes place over error-free channels, the report is equal to the local decision with probability 1 for honest nodes and with probability $1 - P_{\text{mal}}$ for Byzantines. Hence, according to the local decision error model, the probabilities of the reports for honest nodes are derived as:

$$p\left(r_{ij}|s_j, a_i = 1\right) = (1 - \varepsilon)\delta_{(r_{ij} - s_j)} + \varepsilon(1 - \delta_{(r_{ij} - s_j)}), \tag{6.1}$$

where $\delta_{(a)}$ is defined as:

$$\delta_{(a)} = \begin{cases} 1, & \text{if } a = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{6.2}$$

On the other hand, by introducing $\delta = \varepsilon(1 - P_{\text{mal}}) + (1 - \varepsilon)P_{\text{mal}}$, i.e.,

the probability that the FC receives a wrong report from a byzantine node is given by:

$$p\left(r_{ij}|s_j, a_i = 0\right) = (1-\delta)\delta_{(r_{ij}-s_j)} + \delta(1-\delta_{(r_{ij}-s_j)}). \tag{6.3}$$

As for the number of Byzantines, it is assumed that the states of the nodes are independent of each other and the state of each node is described by a Bernoulli random variable with parameter α, that is $p(a_i = 0) = \alpha, \forall i$. In this way, the number of byzantine nodes in the network is a random variable following a binomial distribution, corresponding to the constrained maximum entropy in Chapter 5 with $p\left(a^n\right) = \prod_i p(a_i)$, where $p(a_i) = \alpha(1-a_i)+(1-\alpha)a_i$.

Regarding the sequence of states s^m, we consider a Markov model as shown in Figure 6.1 , i.e., $p\left(s^m\right) = \prod_j p(s_j|s_{j-1})$. The transition probabilities are given by $p(s_j|s_{j-1}) = 1-\rho$ if $s_j = s_{j-1}$ and $p(s_j|s_{j-1}) = \rho$ when $s_j \neq s_{j-1}$, whereas for $j = 1$ we have $p(s_1|s_0) = p(s_1) = 0.5$.

In this chapter we look for the *bitwise* Maximum A Posteriori Probability (MAP) estimation of the system states $\{s_j\}$ which reads as follows:

$$\begin{aligned}
\hat{s}_j &= \arg \max_{s_j \in \{0,1\}} \; p\left(s_j|\mathbf{R}\right) \\
&= \arg \max_{s_j \in \{0,1\}} \sum_{\{s^m, a^n\} \setminus s_j} p\left(s^m, a^n|\mathbf{R}\right) \quad \text{(law of total probability)} \\
&= \arg \max_{s_j \in \{0,1\}} \sum_{\{s^m, a^n\} \setminus s_j} p\left(\mathbf{R}|s^m, a^n\right) p(s^m)p(a^n) \quad \text{(Bayes)} \\
&= \arg \max_{s_j \in \{0,1\}} \sum_{\{s^m, a^n\} \setminus s_j} \prod_{ij} p\left(r_{ij}|s_j, a_i\right) \prod_j p(s_j|s_{j-1}) \prod_i p(a_i)
\end{aligned} \tag{6.4}$$

where the notation $\sum_{\setminus}$ denotes a summation over all the possible combinations of values that the variables contained in the expression within the summation may assume, by keeping the parameter indicated after the operator $\setminus$ fixed. The optimization problem in (6.4) has been solved in the previous chapter for the case of independent system states. Even in such a simple

case, however, the complexity of the optimum decision rule is exceedingly large, thus limiting the use of the optimum decision only in the case of small observation windows (typically m not larger than 10). In the next section we present a sub-optimum solution of (6.4) based on message Passing, which greatly reduces the computational complexity at the price of a negligible loss of accuracy.

6.3 A Decision Fusion Algorithm Based on Message Passing

6.3.1 Introduction to Sum-product Message Passing

In this section we provide a brief introduction to the Message Passing (MP) algorithm for marginalization of sum-product problems. Let us start by considering N binary variables $\mathbf{z} = \{z_1, z_2, \ldots, z_N\}$, $z_i \in \{0, 1\}$. Then, consider the function $f(\mathbf{z})$ with factorization:

$$f(\mathbf{z}) = \prod_k f_k(\mathcal{Z}_k) \tag{6.5}$$

where f_k, $k = 1, \ldots, M$ are functions of a subset $\mathcal{Z}_k$ of the whole set of variables. We are interested in computing the marginal of f with respect to a general variable z_i, defined as the sum of f over all possible values of $\mathbf{z}$, i.e.:

$$\mu(z_i) = \sum_{\mathbf{z} \setminus z_i} \prod_k f_k(\mathcal{Z}_k) \tag{6.6}$$

where notation $\sum_{\mathbf{z} \setminus z_i}$ denotes a sum over all possible combinations of values of the variables in $\mathbf{z}$ by keeping z_i fixed. We are interested in finding the value z_i that optimizes Equation (6.6). Note that marginalization problems occur when we want to compute any arbitrary probability from joint probabilities by summing out variables we are not interested in. In this general setting, since the vector $\mathbf{z}$ has N binary elements, determining the marginals by exhaustive search requires 2^N operations. However, in many situations it is possible to exploit the distributive law of multiplication to get a substantial reduction in complexity.

To elaborate, let us associate with problem (6.6) a bipartite *factor graph*, in which for each variable we draw a variable node (circle) and for each factor we draw a factor node (square). A variable node is connected to a factor node k by an edge if and only if the corresponding variable belongs to $\mathcal{Z}_k$. This means that the set of vertices is partitioned into two groups (the set of nodes corresponding to variables and the set of nodes corresponding to factors) and that an edge always connects a variable node to a factor node.
Let us now assume that the factor graph is a single tree, i.e., a graph in which any two nodes are connected by exactly one path. Under this assumption, the message passing algorithm is able to come out with the exact marginalization after a single iteration. Instead, when the graph is not a tree, i.e., it contains cycles, the algorithm does not give the exact calculation, and it is in general observed that better solutions are obtaied by iterating, even if there is no guarantee that the algorithm converges or that it gives a close-to-the-optimun solution. In general, the longer the cycles the closer the graph to a tree, i.e., the better the message passing solution is. In the case of a tree, it is straightforward to derive an algorithm which allows to solve the marginalization problem with reduced complexity. The algorithm is the MP algorithm, which has been broadly used in the last years in channel coding applications [130], [131].

To describe how the MP algorithm works, let us first define messages as 2-dimensional vectors with binary elements, denoted by $\mathbf{m} = \{m(0), m(1)\}$. Such messages are exchanged between variable nodes and factor nodes and viceversa, according to the following rules. Let us first consider variable-to-factor messages ($\mathbf{m}_{vf}$), and take the portion of factor graph depicted in Figure 6.2 as an illustrative example. In this graph, the variable node z_i is connected to L factor nodes, namely $f_1, f_2, \ldots, f_L$. For the MP algorithm to work properly, node z_i must deliver the messages $\mathbf{m}_{vf}^{(l)}$, $l = 1, \ldots, L$ to all its adjacent nodes. Without loss of generality, let us focus on message $\mathbf{m}_{vf}^{(1)}$. Such a message can be evaluated and delivered upon receiving messages $\mathbf{m}_{fv}^{(l)}$, $l = 2, \ldots, L$, i.e., upon receiving messages from all function nodes except f_1. In particular, $\mathbf{m}_{vf}^{(1)}$ may be straightforwardly evaluated by calculating the

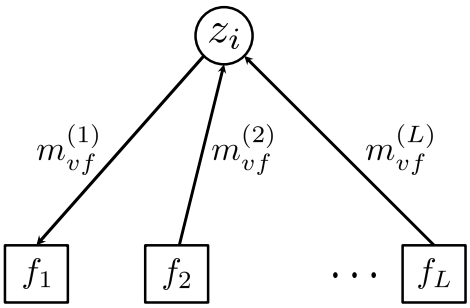


Figure 6.2: *Node-to-factor message passing.*

element-wise product of the incoming messages, i.e.:

$$m_{vf}^{(1)}(q) = \prod_{j=2}^{L} m_{fv}^{(j)}(q), \tag{6.7}$$

for $q = 0, 1$. Let us now consider factor-to-variable messages, and refer to the factor graph of Figure 6.3 where P variable nodes are connected to the factor node f_k, i.e., according to the previous notation, $\mathcal{Z}_k = \{z_1, \ldots, z_P\}$. In this case, the node f_k must deliver the messages $\mathbf{m}_{fv}^{(l)}$, $l = 1, \ldots, P$ to all its adjacent nodes. Let us consider again $\mathbf{m}_{fv}^{(1)}$: upon receiving the messages $\mathbf{m}_{vf}^{(l)}$, $l = 2, \ldots, P$, f_k may evaluate the message $\mathbf{m}_{fv}^{(1)}$ as:

$$m_{fv}^{(1)}(q) = \sum_{z_2,\ldots,z_P} \left[f_k\left(q, z_2, \ldots, z_P\right) \prod_{p=2}^{P} m_{vf}^{(p)}(z_p) \right] \tag{6.8}$$

for $q = 0, 1$.

Given the message passing rules at each node, it is possible to derive the MP algorithm which allows to compute the marginals in Equation (6.6). The process starts at the leaf nodes, i.e., those nodes which have only one connecting edge. In particular, each variable leaf node passes an all-ones message

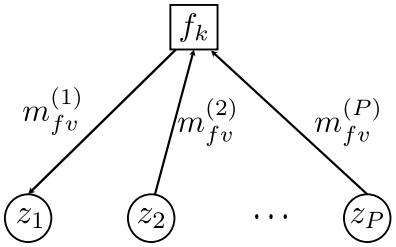


Figure 6.3: *Factor-to-node message passing.*

to its adjacent factor node, whilst each factor leaf node, say $f_k(z_i)$ passes the message $m_{fv}^{(k)}(q) = f_k(z_i = q)$ to its adjacent node z_i. After initialization at leaf nodes, for every edge we can compute the outgoing message as soon as all incoming messages from all other edges connected to the same node are received (according to the message passing rules (Equation 6.7 and Equation 6.8). When a message has been sent in both directions along every edge the algorithm stops. This situation is depicted in Figure 6.4: upon receiving messages from all its adjacent factor nodes, node z_i can evaluate the exact marginal as:

$$\mu(z_i) = \prod_{k=1,\ldots,L} m_{fv}^{(k)}(z_i). \tag{6.9}$$

With regard to complexity, factors to variables message passing can be accomplished with 2^P operations, P being the number of variables in f_k. On the other hand, variables to nodes message passing's complexity can be neglected, and, hence, the MP algorithm allows to noticeably reduce the complexity of the problem provided that the numerosity of $\mathcal{Z}_k$ is much lower than N. With regard to the optimization, Equation 6.9 evaluates the marginal for both $z_i = 0$ and $z_i = 1$, which represent the approximated computation of the sum-product for both hypotheses. Hence, the optimization is obtained by choosing the value of z_i which maximizes it.

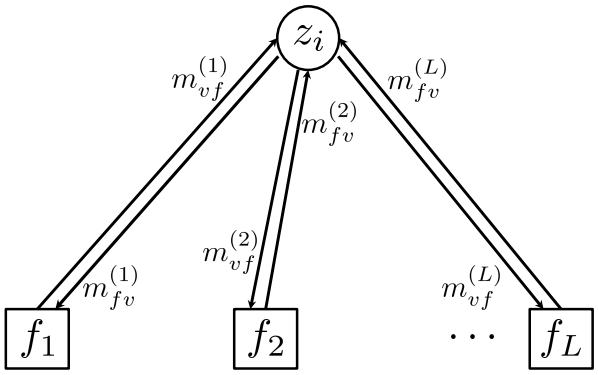


Figure 6.4: *End of message passing for node z_i.*

6.3.2 Nearly-optimal data fusion by means of message passing

The objective function of the optimal fusion rule expressed in (6.4) can be seen as a marginalization of a sum product of functions of binary variables, and, as such, it falls within the MP framework described in the previous Section. More specifically, the variables are the system states s_j and the status of the nodes a_i, while the functions are the probabilities of the reports shown in Equations 6.1 and 6.3, the conditional probabilities $p(s_j|s_{j-1})$, and the a-priori probabilities $p(a_i)$. The resulting bipartite graph is shown in Figure 6.5.

It is worth noting that the graph is a loopy graph, i.e., it contains cycles, and as such it is not a tree. However, although it was originally designed for acyclic graphical models, it was found that the MP algorithm can be used for general graphs, e.g., in channel decoding problems [132]. In general, when the marginalization problem is associated to a loopy graph, the implementation of MP requires to establish a scheduling policy to initiate the procedure, so that variable nodes may receive messages from all the connected factors and evaluate the marginals. In this case, a single run of the MP algorithm may not be sufficient to achieve a good approximation of the exact marginals, and progressive refinements must be obtained through successive iterations.

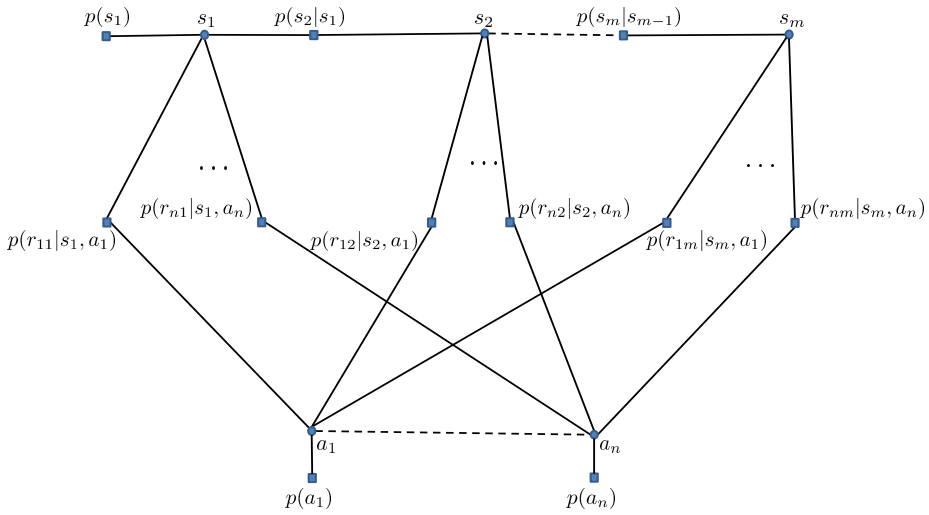


Figure 6.5: *Factor graph for the problem at hand.*

However, in the presence of loopy graphs, there is no guarantee of either convergence or optimality of the final solution. In many cases, the performance of the message-passing algorithms is closely related to the structure of the graph, in general, and its cycles, in particular. In the literature of the field of channel coding, e.g., see [133], researches reached the conclusion that, for good performance, the factor graph should not contain short cycles. In our case, it is possible to see from Figure 6.5 that the shortest cycles have order 6, i.e., a message before returning to the sender must cross at least six different nodes. We speculate that such a minimum cycles length is sufficient to provide good performance for the problem at hand. We will prove through simulations that such a conjecture is true.

To elaborate further, based on the graph of Figure 6.5 and on the general MP rules reported in the previous section, we are now capable of deriving the messages for the scenario at hand. In Figure 6.6, we display all the messages for the graph in Figure 6.5 that are exchanged to estimate in parallel each of

the states $s_{j,j}$ in the vector $s^m = \{s_1, s_2, \ldots, s_m\}$. Specifically, we have:

$$
\begin{array}{lcll}
\tau_j^{(l)}(s_j) & = & \varphi_j^{(l)}(s_j) \prod_{i=1}^{n} \nu_{ij}^{(u)}(s_j) & j = 1, \ldots, m \\
\tau_j^{(r)}(s_j) & = & \varphi_j^{(r)}(s_j) \prod_{i=1}^{n} \nu_{ij}^{(u)}(s_j) & j = 1, \ldots, m \\
\varphi_j^{(l)}(s_j) & = & \sum_{s_{j+1}=0,1} p\left(s_{j+1}|s_j\right) \tau_{j+1}^{(l)}(s_{j+1}) & j = 1, \ldots, m-1 \\
\varphi_j^{(r)}(s_j) & = & \sum_{s_{j-1}=0,1} p\left(s_j|s_{j-1}\right) \tau_{j-1}^{(r)}(s_{j-1}) & j = 2, \ldots, m \\
\varphi_1^{(r)}(s_1) & = & p(s_1) & \\
\nu_{ij}^{(u)}(s_j) & = & \sum_{a_i=0,1} p\left(r_{ij}\,|s_j, a_i\right) \lambda_{ji}^{(u)}(a_i) & j = 1, \ldots, m, \quad i = 1, \ldots, n \\
\nu_{ij}^{(d)}(s_j) & = & \varphi_j^{(r)}(s_j)\varphi_j^{(l)}(s_j) \prod_{\substack{k=1 \\ k \neq i}}^{n} \nu_{kj}^{(u)}(s_j) & j = 1, \ldots, m-1, \quad i = 1, \ldots, n \\
\nu_{im}^{(d)}(s_m) & = & \varphi_j^{(r)}(s_m) \prod_{\substack{k=1 \\ k \neq i}}^{n} \nu_{km}^{(u)}(s_m) & i = 1, \ldots, n \\
\lambda_{ji}^{(d)}(a_i) & = & \sum_{s_j=0,1} p\left(r_{ij}\,|s_j, a_i\right) \nu_{ij}^{(d)}(s_j) & j = 1, \ldots, m, \quad i = 1, \ldots, n \\
\lambda_{ji}^{(u)}(a_i) & = & \omega_i^{(u)}(a_i) \prod_{\substack{q=1 \\ q \neq j}}^{m} \lambda_{qi}^{(d)}(a_i) & j = 1, \ldots, m, \quad i = 1, \ldots, n \\
\omega_i^{(d)}(a_i) & = & \prod_{j=1}^{m} \lambda_{ji}^{(d)}(a_i) & i = 1, \ldots, n \\
\omega_i^{(u)}(a_i) & = & p(a_i) & i = 1, \ldots, n
\end{array}
\tag{6.10}
$$

As for the scheduling policy, the MP procedure is initiated by sending the messages $\lambda_{ji}^{(u)}(a_i) = \omega_i^{(u)}(a_i)$ to all $p\left(r_{ij}\,|s_j, a_j\right)$ factor nodes, and by sending the message $p(s_1)$ to the variable node s_1. Hence, the MP proceeds according to the general message passing rules, until all variable nodes are able to compute the respective marginals. When this happens, the first iteration is concluded. Then, successive iterations are carried out by starting from leaf nodes and by taking into account the messages received at the previous iteration for the evaluation of new messages. Hence, the algorithm is stopped upon achieving convergence of messages, or after a maximum number of iterations.

The MP scheme described above can be simplified by observing that messages can be normalized without affecting the normalized marginals. Henceforward, let us consider as normalization factors the sum of the elements of

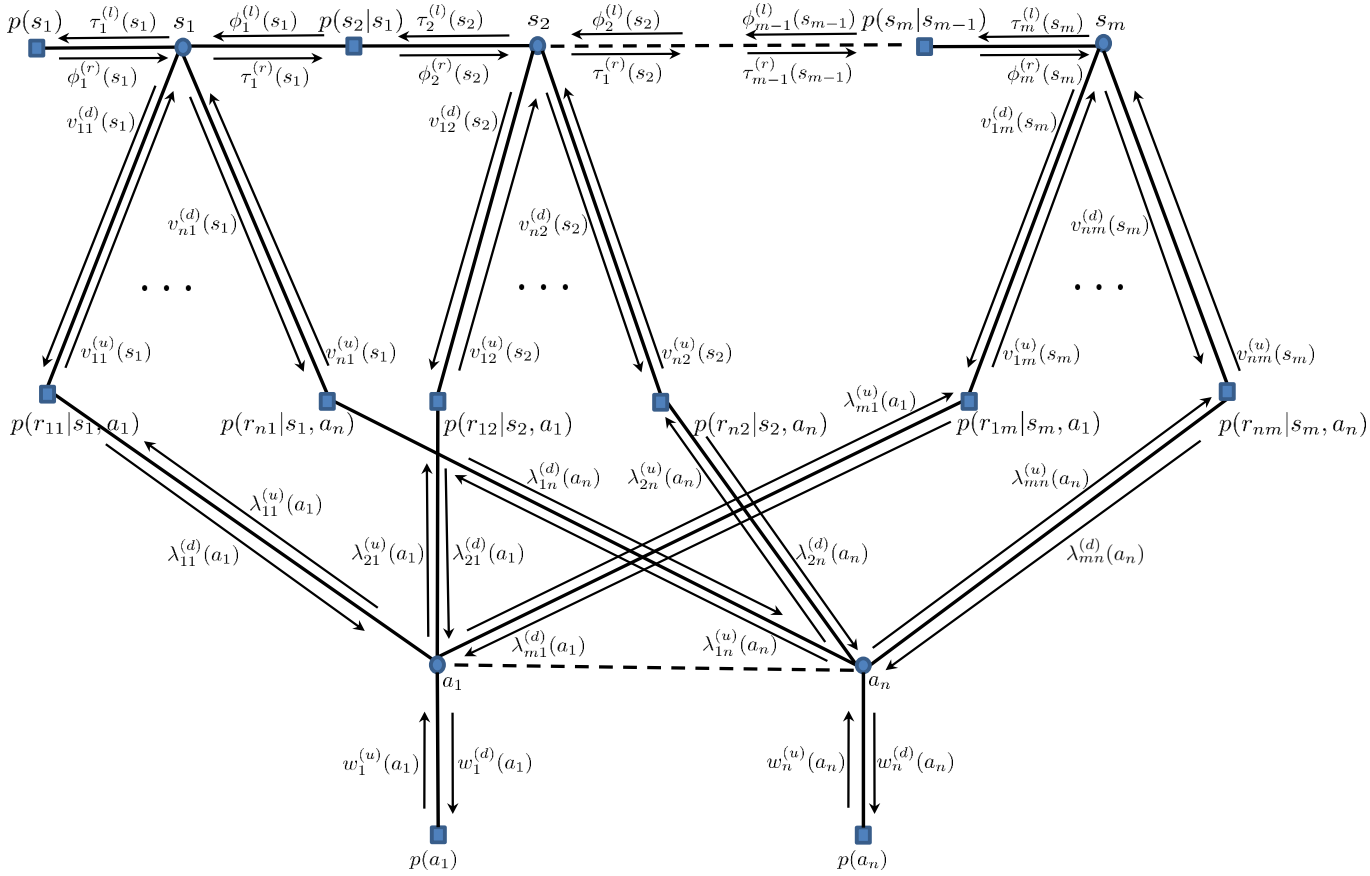


Figure 6.6: Factor graph for the problem at hand with the illustration of all the exchanged messages.

the messages, i.e., if we consider for example $\tau_j^{(l)}(s_j)$, the normalization factor is $\tau_j^{(l)}(0) + \tau_j^{(l)}(1)$. In this case, the normalized messages, say $\bar{\tau}_j^{(l)}(s_j)$ can be conveniently represented as scalar terms in the interval $(0, 1)$, e.g., we can consider $\bar{\tau}_j^{(l)}(0)$ only since $\bar{\tau}_j^{(l)}(1) = 1 - \bar{\tau}_j^{(l)}(0)$. Accordingly, the normalized messages can be evaluated as:

$$\bar{\tau}_j^{(l)} = \frac{\bar{\varphi}_j^{(l)} \prod_{i=1}^{n} \bar{\nu}_{ij}^{(u)}}{\bar{\varphi}_j^{(l)} \prod_{i=1}^{n} \bar{\nu}_{ij}^{(u)} + (1 - \bar{\varphi}_j^{(l)}) \prod_{i=1}^{n} (1 - \bar{\nu}_{ij}^{(u)})} \quad j = 1, \ldots, m$$

$$\bar{\tau}_j^{(r)} = \frac{\bar{\varphi}_j^{(r)} \prod_{i=1}^{n} \bar{\nu}_{ij}^{(u)}}{\bar{\varphi}_j^{(r)} \prod_{i=1}^{n} \bar{\nu}_{ij}^{(u)} + (1 - \bar{\varphi}_j^{(r)}) \prod_{i=1}^{n} (1 - \bar{\nu}_{ij}^{(u)})} \quad j = 1, \ldots, m$$

$$\bar{\varphi}_j^{(l)} = \rho \bar{\tau}_{j+1}^{(l)} + (1 - \rho)(1 - \bar{\tau}_{j+1}^{(l)}) \quad j = 1, \ldots, m-1$$

$$\bar{\varphi}_j^{(r)} = \rho \bar{\tau}_{j-1}^{(r)} + (1 - \rho)(1 - \bar{\tau}_{j-1}^{(r)}) \quad j = 2, \ldots, m$$

$$\bar{\varphi}_1^{(r)} = p(s_1 = 0)$$

$$\bar{\nu}_{ij}^{(u)} = \frac{p\,(r_{ij}\,|0,0\,)\,\bar{\lambda}_{ji}^{(u)} + p\,(r_{ij}\,|0,1\,)\,(1 - \bar{\lambda}_{ji}^{(u)})}{\kappa_1 + \kappa_2}$$

where, $\kappa_1 = p\,(r_{ij}\,|0,0\,)\,\bar{\lambda}_{ji}^{(u)} + p\,(r_{ij}\,|0,1\,)\,(1 - \bar{\lambda}_{ji}^{(u)})$

and $\kappa_2 = p\,(r_{ij}\,|1,0\,)\,\bar{\lambda}_{ji}^{(u)} + p\,(r_{ij}\,|1,1\,)\,(1 - \bar{\lambda}_{ji}^{(u)})$

$j = 1, \ldots, m, \quad i = 1, \ldots, n$

$$\bar{\nu}_{ij}^{(d)} = \frac{\bar{\varphi}_j^{(r)} \bar{\varphi}_j^{(l)} \prod_{\substack{k=1 \\ k \neq j}}^{n} \bar{\nu}_{ki}^{(u)}}{\bar{\varphi}_j^{(r)} \bar{\varphi}_j^{(l)} \prod_{\substack{k=1 \\ k \neq i}}^{n} \bar{\nu}_{ki}^{(u)} + (1 - \bar{\varphi}_j^{(r)})(1 - \bar{\varphi}_j^{(l)}) \prod_{\substack{k=1 \\ k \neq i}}^{n} (1 - \bar{\nu}_{ki}^{(u)})}$$

$j = 1, \ldots, m-1, \quad i = 1, \ldots, n$

$$
\begin{aligned}
\bar{\nu}_{jm}^{(d)} =& \frac{\bar{\varphi}_m^{(r)} \prod\limits_{\substack{k=1 \\ k\neq i}}^{n} \bar{\nu}_{km}^{(u)}}{\bar{\varphi}_m^{(r)} \prod\limits_{\substack{k=1 \\ k\neq i}}^{n} \bar{\nu}_{km}^{(u)} + (1-\bar{\varphi}_m^{(r)}) \prod\limits_{\substack{k=1 \\ k\neq i}}^{n} (1-\bar{\nu}_{km}^{(u)})} \quad i = 1, \ldots, n \\
\bar{\lambda}_{ji}^{(d)} =& \frac{p\left(r_{ij}\,|0,0\right) \bar{\nu}_{ij}^{(d)} + p\left(r_{ij}\,|1,0\right)(1-\bar{\nu}_{ij}^{(d)})}{\tau_1 + \tau_2} \\
& \text{where,} \quad \tau_1 = p\left(r_{ij}\,|0,0\right) \bar{\nu}_{ij}^{(d)} + p\left(r_{ij}\,|1,0\right)(1-\bar{\nu}_{ij}^{(d)}) \\
& \text{and} \quad \tau_2 = p\left(r_{ij}\,|0,1\right) \bar{\nu}_{ij}^{(d)} + p\left(r_{ij}\,|1,1\right)(1-\bar{\nu}_{ij}^{(d)}) \\
& j = 1, \ldots, m, \quad i = 1, \ldots, n \\
\bar{\lambda}_{ji}^{(u)} =& \frac{\bar{\omega}_i^{(u)} \prod\limits_{\substack{q=1 \\ q\neq j}}^{m} \bar{\lambda}_{qi}^{(d)}}{\bar{\omega}_i^{(u)} \prod\limits_{\substack{q=1 \\ q\neq j}}^{m} \bar{\lambda}_{qi}^{(d)} + (1-\bar{\omega}_i^{(u)}) \prod\limits_{\substack{q=1 \\ q\neq j}}^{m} (1-\bar{\lambda}_{qi}^{(d)})} \\
& j = 1, \ldots, m, \quad i = 1, \ldots, n \\
\bar{\omega}_i^{(d)} =& \frac{\prod\limits_{j=1}^{m} \bar{\lambda}_{ji}^{(d)}}{\prod\limits_{j=1}^{m} \bar{\lambda}_{ji}^{(d)} + \prod\limits_{j=1}^{m} (1-\bar{\lambda}_{ji}^{(d)})} \quad i = 1, \ldots, n \\
\bar{\omega}_i^{(u)} =& p(a_i = 0) \quad i = 1, \ldots, n
\end{aligned}
\tag{6.11}
$$

6.4 Simulation Results and Discussion

In this section, we present an analysis of the performance of the MP decision fusion algorithm. We first consider the computational complexity, then we pass to the performance evaluation in terms of error probability. In particular, the performance of the MP-based scheme is compared to the optimum fusion rule in Chapter 5 (whenever possible), the soft and the hard isolation schemes presented in Chapter 4 and the simple majority rule. In this comparison, both independent and Markovian system states are considered, for both small and large observation windows m.

6.4.1 Complexity Assessment

In order to evaluate the complexity of the message passing algorithm and compare it to that of the optimum fusion scheme, both the number of operations and the running time are considered. The number of operations means the number of additions, substractions, multiplications and divisions performed by the algorithm to estimate the vector of system states s^m.

By looking at Equation 6.11, it can be seen that running the message passing algorithm requires the following number of operations:

- $3n + 5$ operations for each of $\bar{\tau}_j^{(l)}$ and $\bar{\tau}_j^{(r)}$.
- 3 operations for each of $\bar{\varphi}_j^{(l)}$ and $\bar{\varphi}_j^{(r)}$.
- 11 operations for $\bar{\nu}_{ij}^{(u)}$.
- $3n + 5$ operations for $\bar{\nu}_{ij}^{(d)}$.
- $3n + 2$ operations for $\bar{\nu}_{im}^{(d)}$.
- 11 operations for $\bar{\lambda}_{ji}^{(d)}$.
- $3m + 2$ operations for each of $\bar{\lambda}_{ji}^{(u)}$ and $\bar{\omega}_i^{(d)}$.

summing up to $12n + 6m + 49$ operations for each iteration over the factor graph. On the other hand, in the case of independent node states, the optimal scheme in Chapter 5 requires $2^m(m + n)$ operations. Therefore, the MP algorithm is much less computationally expensive since it passes from an exponential to a linear complexity in m. An example of the difference in computational complexity between the optimum and the MP algorithms is depicted in Figure 6.7 for fixed m and in Figure 6.8 for fixed n.

With regard to time complexity, Table 6.1 reports the running time of the MP and the optimal schemes. For $n = 20$, the optimal scheme running time is 17.547 times larger than that of the message passing algorithm. On the other hand, for the case of $n = 100$, the optimal scheme needs around 4.258 times more than the message passing scheme. The tests have been conducted using Matlab 2014b running on a machine with 64-bit windows 7 OS with 16,0 GB of installed RAM and Intel Core i7-2600 CPU @ 3.40GHz.

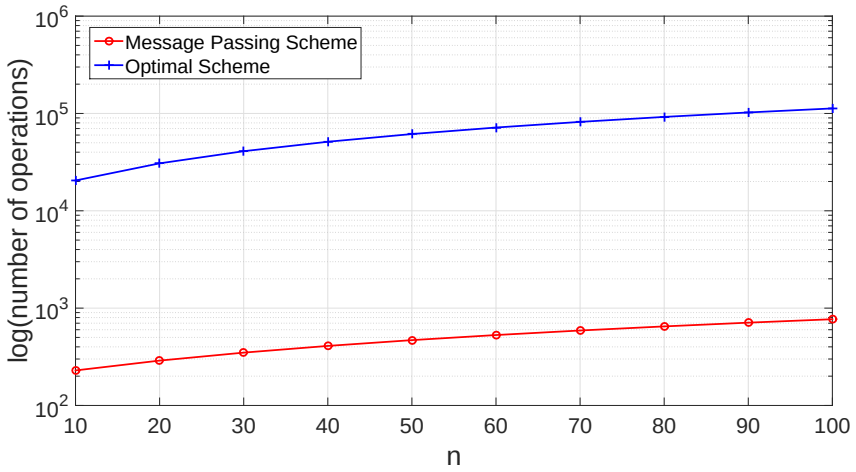


Figure 6.7: *Number of operations required for different n, $m = 10$ and 5 message passing local iterations for message passing and optimal schemes.*

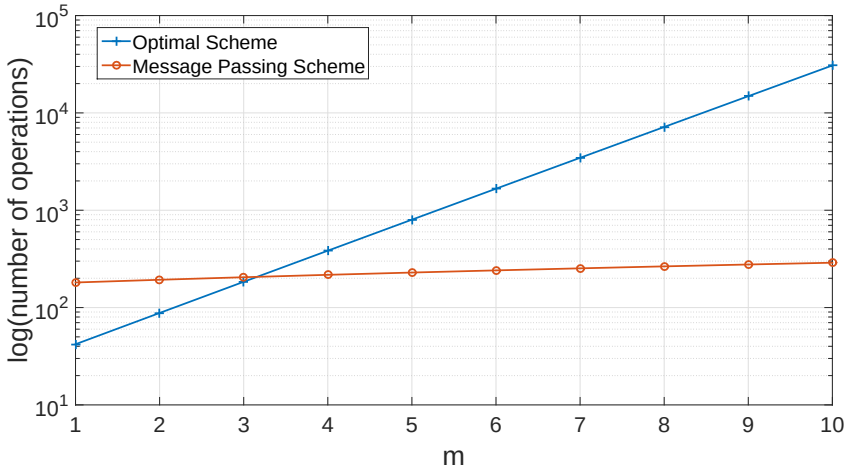


Figure 6.8: *Number of operations required for different m, $n = 20$ and 5 message passing local iterations for message passing and optimal schemes.*

Setting/Scheme	Message Passing	Optimal
$n = 20$,$\alpha = 0.45$	943.807114	1.6561e+04
$n = 100$,$\alpha = 0.49$	4888.821497	2.0817e+04

Table 6.1: *Running Time (in seconds) for the Optimal and the Message Passing Algorithms for:* $m = 10$, $\varepsilon = 0.15$, *Number of Trials* $= 10^5$ *and Message Passing Iterations* $= 5$.

6.4.2 Performance Evaluation

In this section, we present a set of numerical simulations to evaluate the performance of the message passing algorithm and compare them to those obtained by other schemes. The results are divided into four parts. The first two parts consider, respectively, simulations performed with small and large observation windows m. Then, the third part used to investigate the optimum behaviour of the Byzantines over a range of observation windows size. Finally, in the last part, a comparison between the case of independent and Markovian system states is presented.

The simulations were carried out according to the following setup. A network with $n = 20$ nodes is considered, $\varepsilon = 0.15$, $\rho = \{0.95, 0.5\}$ corresponding to Markovian and independent sequence of system states, respectively. The probability α that a node is Byzantine is in the range $[0, 0.45]$ corresponding to a number of Byzantines between 0 and 9. As to P_{mal} it is set to either 0.5 or 1[1]. The number of message passing iterations is 5. For each setting, we estimated the error probability over 10^5 trials.

6.4.2.1 Small m BT: 'Small/Short observation window'

We start by considering a small observation window, namely $m = 10$. With such a small value of m, in fact, it is possible to compare the performance of the message passing algorithm to that of the optimum decision fusion rule. The results we obtained are reported in Figure 6.9. Upon inspection of the figure, the message passing algorithm outperforms the Majority, Soft and

[1]It is know from Chapter 5 that for the Byzantines the optimum choice of P_{mal} is either 0.5 or 1 depending on the considered setup.

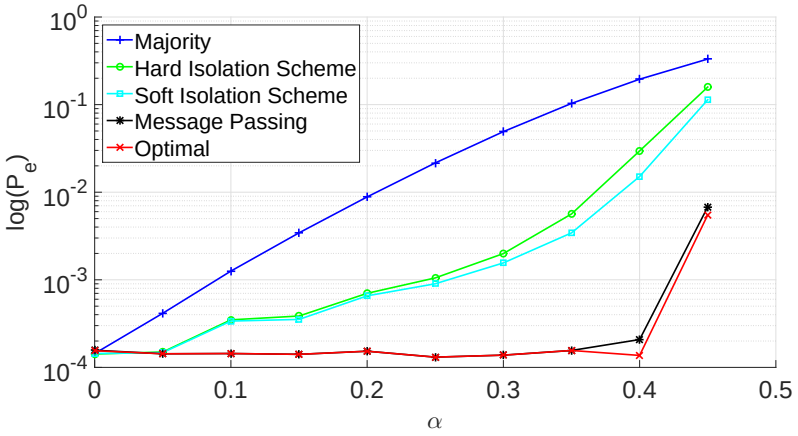


Figure 6.9: *Error probability as a function of* α *for the following setting:* $n = 20$, *independent Sequence of States* $\rho = 0.5$, $\varepsilon = 0.15$, $m = 10$ *and* $P_{\text{mal}} = 1.0$.

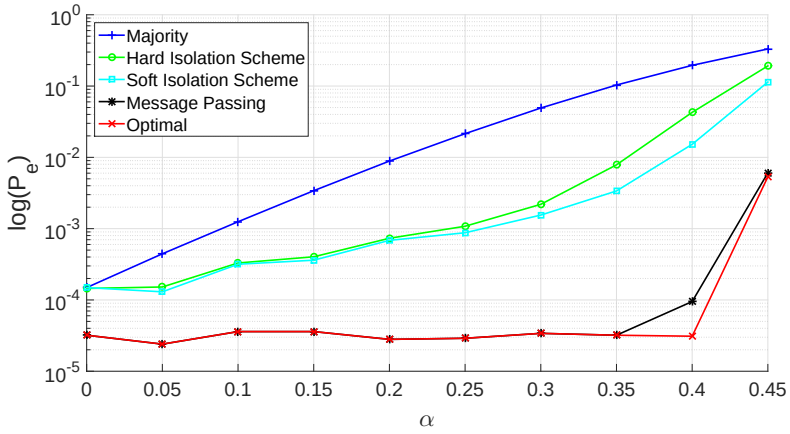


Figure 6.10: *Error probability as a function of* α *for the following setting:* $n = 20$, *Markovian Sequence of States* $\rho = 0.95$, $\varepsilon = 0.15$, $m = 10$ *and* $P_{\text{mal}} = 1.0$

Hard isolation schemes. More interestingly, the message passing algorithm gives nearly optimal performance, with only a negligible performance loss with respect to the optimum scheme.

Figure 6.10 confirms the results shown in Figure 6.9 for Markovian system states ($\rho = 0.95$).

6.4.2.2 Large m BT: Large observation window'

Having shown the near optimality of the message passing fusion algorithm for small values of m, we now leverage on the small computational complexity of such a scheme to evaluate its performance for large values of m. in particular, for the simulations, we set $m = 30$. As shown in Figure 6.11, by increasing the observation window all the schemes give better performance, with the message passing algorithm always providing the best performance. Interestingly, in this case, when the attacker uses $P_{\text{mal}} = 1.0$, the message passing algorithm permits to almost nullify the attack of the Byzantines for all the values of α. The reason is that, using $P_{\text{mal}} = 1.0$ conveys more information to the FC about the Byzantines and consequently, makes their detection easier. Concerning the residual error probability, it is due to the fact that, even when there are no Byzantines in the network ($\alpha = 0$), there is still an error floor caused by the local errors at the nodes ε. For the case of independent states, such an error floor is around 10^{-4}. In Figure 6.11 and 6.12, this error floor decreases to about 10^{-5} because of the additional a-priori information available in the Markovian case.

6.4.2.3 Optimal choice of P_{mal} for the Byzantines

One of the main results derived from the previous chapter, is that setting $P_{\text{mal}} = 1$ is not necessarily the optimal choice for the Byzantines. In fact, when the FC manages to identify which are the malicious nodes, it can exploit the fact the malicious nodes always flip the result of the local decision to get useful information about the system state. In such cases, it is preferable for the Byzantines to use $P_{\text{mal}} = 0.5$ since in this way the reports send to the FC does not convey any information about the status of the system. However, in Chapter 5, it was not possible to derive exactly the limits determining the two different behaviours for the Byzantines due to the impossibility of applying

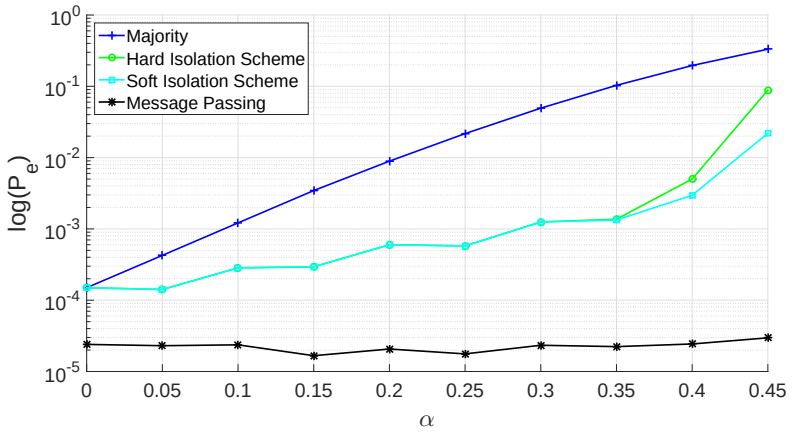


Figure 6.11: *Error probability as a function of* α *for the following setting:* $n = 20$, *Markovian Sequence of States* $\rho = 0.95$, $\varepsilon = 0.15$, $m = 30$ *and* $P_{\text{mal}} = 1.0$.

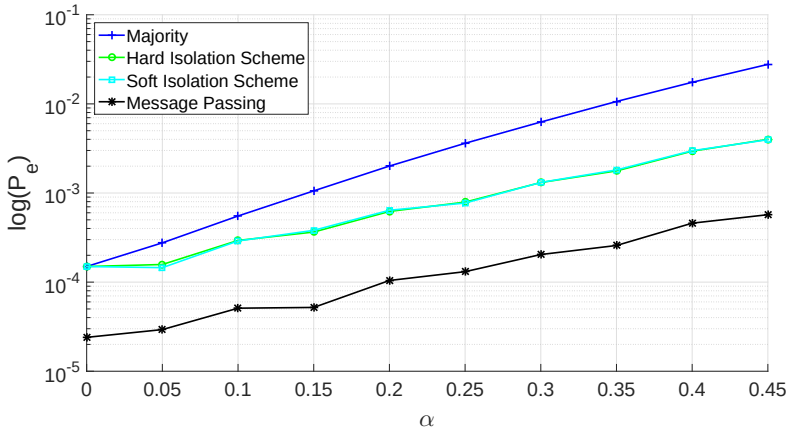


Figure 6.12: *Error probability as a function of* α *for the following setting:* $n = 20$, *Markovian Sequence of States* $\rho = 0.95$, $\varepsilon = 0.15$, $m = 30$ *and* $P_{\text{mal}} = 0.5$.

the optimum algorithm in conjunction with large observation windows. By exploiting the low complexity of the message passing scheme, we are now able to overcome such limitation.

Specifically, we present an additional set of experiments by fixing $\alpha = 0.45$ and varying the observation window in the interval [8,20]. The results obtained confirm the general behaviour observed in Chapter 5. For instance, in Figure 6.13, $P_{\text{mal}} = 1.0$ remains the Byzantines' optimal choice up to $m = 13$, while for $m > 13$, it is preferable for them to use $P_{\text{mal}} = 0.5$. Similar results are obtained for independent system states as shown in Figure 6.14.

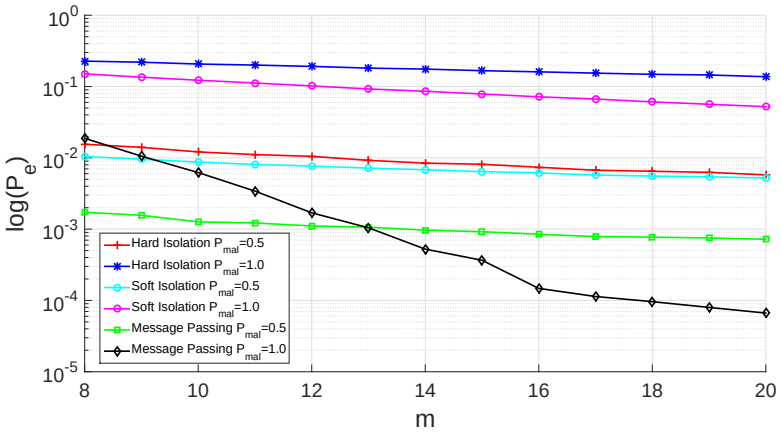


Figure 6.13: *Error probability as a function of m for the following settings: $n = 20$, Markovian Sequence of States $\rho = 0.95$, $\varepsilon = 0.15$ and $\alpha = 0.45$.*

6.4.2.4 Comparison between independent and Markovian System States

In this part, we compared the cases of Markovian system states and the one of independent system states.

By looking at Figure 6.13 and 6.14, we see that the Byzantines switch their strategy from $P_{\text{mal}} = 1$ to $P_{\text{mal}} = 0.5$ for a smaller observation window ($m = 10$) in the case of independent states (the switching value for the Markovian case is $m = 13$). This behaviour can be explained by observing that in the

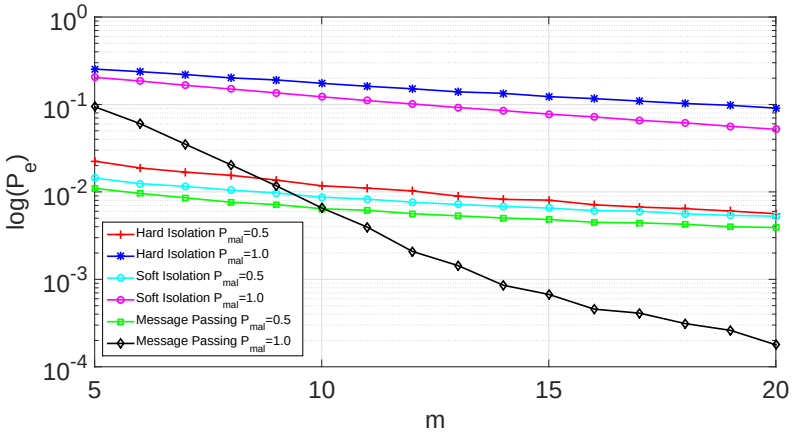


Figure 6.14: *Error probability as a function of* m *for the following settings:* $n = 20$*, independent Sequence of States* $\rho = 0.5$*,* $\varepsilon = 0.15$ *and* $\alpha = 0.45$*.*

case of Markovian states, using $P_{\text{mal}} = 0.5$ results in a strong deviation from the Markovianity assumption of the reports sent to the FC thus making it easier the isolation of byzantine nodes. This is not the case with $P_{\text{mal}} = 1$, since, due to the symmetry of the adopted Markov model, such a value does not alter the expected statistics of the reports.

As a last result, in Figure 6.15, we show a comparison the error probability for the case of independent and Markov sources. Since the interest is to compare the achievable performance for the two cases, only the performance obtained by the optimum and the message passing algorithms is considered. Upon inspection of the figure, it turns out that the case of independent states is more favourable to the Byzantines than the Markov case. The reason is that the FC may exploit the additional a-priori information available in the Markov case to identify the Byzantines and hence make a better decision. Such effect disappears when α approaches 0.5, since in this case the Byzantines tend to dominate the network. In that case, the Byzantines' reports prevail the pool of reports at the FC and hence, the FC becomes nearly *blind* so that even the additional a-priori information about the Markov model does not offer a great help.

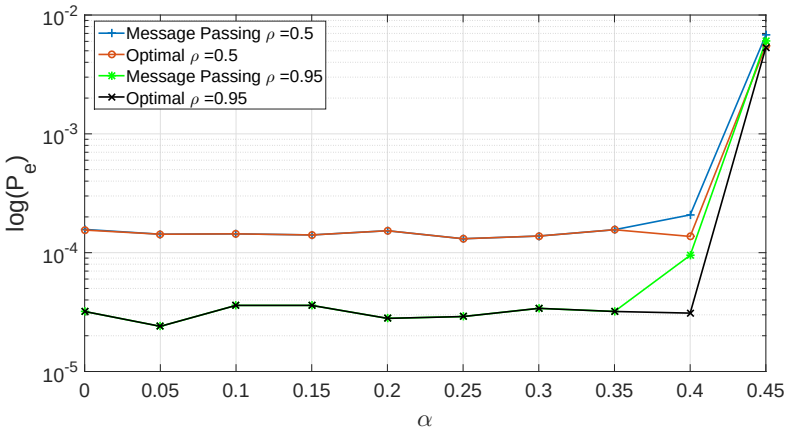


Figure 6.15: *Comparison between the case of independent and Markovian system states ($n = 20$, $\rho = \{0.5, 0.95\}$, $\varepsilon = 0.15$, $m = 10$, $P_{\text{mal}} = 1.0$).*

6.5 Conclusions

In this chapter, we presented a near-optimal message passing algorithm based on factor graph for decision fusion in distributed sensor networks in the presence of Byzantines. The effectiveness of the MP scheme is evaluated by means of extensive numerical simulations both for the case of independent and Markov sequence of states. Experiments showed that, when compared to the optimum fusion scheme, the MP scheme permits to achieve near-optimal performance at a much lower computational cost: specifically, by adopting the new algorithm based on message passing, the complexity is reduced from exponential to linear. Such reduction of the complexity permits to deal with large observation windows, thus further improving the performance of the decision. Results on large observation windows confirmed the dual behavior of the optimum attacking strategy, looking for a trade-off between pushing the FC to make a wrong decision on one hand and reducing the mutual information between the reports and the system state on the other hand. In addition, the experiments showed that the case of independent states is more favorable to Byzantines than the Markovian case, due to the additional a-priori infor-

mation available at the FC in the Markovian case.

Chapter 7

Conclusion

7.1 Introduction

This book provided a game-theoretic approach for adversarial information fusion in distributed sensor networks. We presented several solutions to tackle with the presence of the adversaries in such networks. In this chapter, we summarize the main solutions presented in the book and outline some possible directions for future research.

7.2 Summary

Since information fusion in distributed networks is of great importance in many applications, among them cognitive radio networks, multimedia forensics, wireless sensor networks and many others, securing these networks against attacks and threats became of a crucial and a key enabling factor for their proper functionality. Motivated by this fact, in this book, we studied this problem by considering the possible presence of adversaries that aim at corrupting the functionality of these networks. Following the concepts and resorting to the studying tools typical of adversarial signal processing, a game-theoretic approach is employed to study the security of adversarial information fusion in distributed sensor networks.

In Chapter 2, a review of the basic notions of detection theory and game theory was carried out, while in Chapter 3 we presented an extensive review of the literature concerning the attacks and their mitigation techniques in adversarial distributed sensor networks. Then, in Chapter 4-6 we presented possible solutions to mitigate the effect of the attacks in these networks and using a game-theoretic formulation to study the ultimate performance that can be achieved by the attacker and the defender.

We started by considering an adversarial decision fusion setup in which the nodes send to the FC a vector of binary decisions about the system state. As a heuristic solution, we presented a soft identification and isolation scheme to exclude the reports sent by the adversary, namely the Byzantines, from the decision fusion process. By adopting such a scheme, the FC can assign a reliability value to each node. Then, the competition between the Byzantines and the FC is formalized in a game-theoretic sense and the existence of an equilibrium point for the game is studied. The payoff is computed in terms of the decision error probability when the players play at the equilibrium. By using numerical simulations, we showed that the soft isolation scheme outperforms the common defense mechanisms based on a hard isolation approach.

As a second more rigorous solution, we presented the derivation of the optimum decision fusion rule in the presence of Byzantines in a centralized setup. By observing the system over an observation window, the Maximum A Posteriori Probability (MAP) rule is adopted while assuming that the FC knows the attack strategy of the Byzantines and their distribution across the network. With regard to the knowledge that the FC has about the distribution of Byzantines over the network, many cases were considered. The first scencario considered is the unconstrained maximum entropy in which the uncertainty about the distribution of Byzantines is maximum. Then, we addressed a more favorable scenario to the FC in which the maximum entropy case is subject to a constraint. In this scenario, the FC has more a-priori information about Byzantines's distribution i.e the average or the maximum number of Byzantines in the network. Finally, we investigated the most favorable situation in which the FC knows the exact number of Byzantines present in the network. Concerning the complexity of the optimal fusion rule, we explained an efficient implementation based on Dynamic Programming. Thereafter, a game-theoretic framework is introduced to cope with the lack of knowledge regarding the Byzantines strategy. In such a framework, the FC makes a "guess" by selecting arbitrarily a Byzantine's attacking strategy within the optimum fusion rule. By considering the decision error probability as the payoff, the performance of the FCas well as the Byzantines at the equilibrium for several setups is studied. The main noticeable result of this chapter is that the attacker should follow a mixed strategy Nash equilibrium; this strategy

reaches a trade-off between inducing decision error at the FC and avoiding being caught, by minimizing the mutual information between the information conveyed and the system state. Finally, by comparing the performance of the optimum fusion rule to those of previous works, we showed its superior performance over all the other schemes.

By revisiting the complexity of the optimum fusion rule, as an additional contribution, we presented a near-optimal message passing approach based on factor graph. For this case, we presented a more general model for the observed system in which we examine both independent and Markovian sequences. Then, we showed that the message passing algorithm can give near-optimal performance while reducing the complexity from exponential to linear as a function of the observation window size. In addition, we showed that the case of independent states is more favorable to the Byzantines than the Markovian case, due to the additional a-priori information available at the FC in the Markovian case. Furthermore, based on large observation windows, the dual behavior in the attack strategy of the Byzantines is confirmed.

7.3 Open Issues

There are some avenues for future work on the topics addressed in this book. As a first research direction, the performance of the Byzantines can be improved by giving them more freedom in their strategies. For instance, they can exploit the knowledge of the observation vectors. By exploiting the knowledge of such information, the Byzantines can focus their attack on the most uncertain cases thus avoiding to change the local decision when it is expected that the attack will have no effect on the FC decision. Another approach would be granting the Byzantines the opportunity to coordinate before deciding on the attack and the strategy. Therefore, all the Byzantines can be synchronized together and they can do so by using a pseudo random generator with a common seed to synchronize their local clocks. Considering a scenario where the nodes can send more extensive reports rather than one single bit [134] would be another relevant piece of work.

More interestingly, the concepts of adversarial signal processing can be applied to more complex and fully adaptive networks. The reason is that,

these networks have been playing an increasingly important role in our daily life because of their technological, social, and economical aspects on people's life. Facebook, Twitter, Amazon are examples of these networks. As a first example, on Facebook, the concept of adversarial information fusion can be applied to combat against fake user profiles who generate forged/negative reviews or ratings against a business page and cause financial loss. On Twitter, for instance, we can counter the users or profiles who, by faking information into their tweets, spread falsified information about an event or a trend that can be critical.

Although being aware of the limitations in practical applications, we believe that studying adversarial information fusion can bring an important contribution to the security of networks with distributed sensors.

Bibliography

[1] McAfee, "Net losses: Estimating the global cost of cybercrime," *McAfee, Centre for Strategic & International Studies*, 2014.

[2] Federal bureau of investigation, cyber crime. (2016, September0 26). [Online]. Available: https://www.fbi.gov/investigate/cyber

[3] Official website of the department of homeland security, combating cyber crime. (2016, September0 26). [Online]. Available: https://www.dhs.gov/topic/combating-cyber-crime

[4] A. Marcella Jr and R. S. Greenfield, *Cyber forensics: a field manual for collecting, examining, and preserving evidence of computer crimes.* CRC Press, 2002.

[5] M. Barni and B. Tondi, "The source identification game: An information-theoretic perspective," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 450–463, March 2013.

[6] R. Böhme and M. Kirchner, "Counter-forensics: Attacking image forensics," in *Digital Image Forensics*, H. T. Sencar and N. Memon, Eds. Springer Berlin / Heidelberg, 2012.

[7] L. Pérez-Freire, P. Comesana, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Watermarking security: a survey," in *Transactions on Data Hiding and Multimedia Security I.* Springer, 2006, pp. 41–72.

[8] G. C. Kessler and C. Hosmer, "An overview of steganography," *Advances in Computers*, vol. 83, no. 1, pp. 51–107, 2011.

[9] A. K. Jain, A. Ross, and U. Uludag, "Biometric template security: Challenges and solutions," in *13th European Signal Processing Conference, 2005*, Antalya, Turkey, September 2005, pp. 1–4.

[10] M. Barreno, B. Nelson, A. D. Joseph, and J. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.

[11] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. ACM, 2005, pp. 641–647.

[12] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, Washington, DC, USA, September 2005, pp. 113–126.

[13] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1168–1174, 2008.

[14] M. Barni and F. Pérez-González, "Coping with the enemy: Advances in adversary-aware signal processing," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, May 2013, pp. 8682–8686.

[15] M. Barni and B. Tondi, "Binary hypothesis testing game with training data," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4848–4866, Aug 2014.

[16] M. Barni and B. Tondi, "The security margin: A measure of source distinguishability under adversarial conditions," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2013*, Dec 2013, pp. 225–228.

[17] M. Barni and B. Tondi, "Source distinguishability under distortion-limited attack: An optimal transport perspective," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2145–2159, Oct 2016.

[18] V. Torra and Y. Narukawa, *Modeling decisions: information fusion and aggregation operators*. Springer Science & Business Media, 2007.

[19] J. J. Chao, E. Drakopoulos, and C. C. Lee, "An evidential reasoning approach to distributed multiple-hypothesis detection," in *26th IEEE Conference on Decision and Control*, vol. 26, Los Angeles, California, USA, December 1987, pp. 1826–1831.

[20] N. Xiong and P. Svensson, "Multi-sensor management for information fusion: issues and approaches," *Information fusion*, vol. 3, no. 2, pp. 163–186, 2002.

[21] S. Haykin, D. J. Thomson, and J. H. Reed, "Spectrum sensing for cognitive radio," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 849–877, 2009.

[22] Y. Xiao and F. Hu, *Cognitive radio networks*. CRC press, 2008.

[23] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3386–3407, 2011.

[24] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[25] M. Barni and B. Tondi, "Multiple-observation hypothesis testing under adversarial conditions," in *Proc. of WIFS'13, IEEE International Workshop on Information Forensics and Security*, Guangzhou, China, Nov 2013, pp. 91–96.

[26] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on dempster–shafer theory of evidence," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 593–607, 2013.

[27] Y. L. Sun and Y. Liu, "Security of online reputation systems: The evolution of attacks and defenses." *IEEE Signal Processing Magazine*, vol. 29, no. 2, pp. 87–97, 2012.

[28] D. G. Padmavathi, M. Shanmugapriya *et al.*, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *arXiv preprint arXiv:0909.0576*, 2009.

[29] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[30] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *Commununications Surveys & Tutorials*, vol. 8, no. 2, pp. 2–23, Apr. 2006. [Online]. Available: http://dx.doi.org/10.1109/COMST.2006.315852

[31] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.

[32] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[33] A. Vempaty, T. Lang, and P. Varshney, "Distributed inference with byzantine data: State-of-the-art review on data falsification attacks," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 65–75, Sept 2013.

[34] R. Chen, J. M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, Pheonix AZ, USA, April 2008, pp. 1–5.

[35] R. Chen, J.-M. Park, Y. T. Hou, and J. H. Reed, "Toward secure distributed spectrum sensing in cognitive radio networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 50–55, 2008.

[36] F. R. Yu, H. Tang, M. Huang, Z. Li, and P. C. Mason, "Defense against spectrum sensing data falsification attacks in mobile ad hoc networks with cognitive radios," in *MILCOM 2009, IEEE Military Communications Conference*, Boston, USA, October 2009, pp. 1–7.

[37] H. Modares, R. Salleh, and A. Moravejosharieh, "Overview of security issues in wireless sensor networks," in *2011 Third International Conference on Computational Intelligence, Modelling & Simulation*, Langkawi, Malaysia, September 2011, pp. 308–311.

[38] S. M. Kay, *Fundamentals of statistical signal processing: Detection theory, vol. 2.* Prentice Hall Upper Saddle River, NJ, USA:, 1998.

[39] L. L. Scharf, *Statistical signal processing.* Addison-Wesley Reading, MA, 1991, vol. 98.

[40] J. Mitola, "Cognitive radio—an integrated agent architecture for software defined radio," Ph.D. dissertation, Royal Institute of Technology (KTH), 2000.

[41] V. V. Veeravalli and P. K. Varshney, "Distributed inference in wireless sensor networks," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 100–117, 2012.

[42] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2005.

[43] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: An artificial intelligence approach.* Springer Science & Business Media, 2013.

[44] I. Newton, *The Principia: mathematical principles of natural philosophy.* Univ of California Press, 1999.

[45] P. K. Varshney, *Distributed Detection and Data Fusion.* Springer-Verlag, 1997.

[46] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[47] A. Wald and J. Wolfowitz, "Optimum character of the sequential probability ratio test," *The Annals of Mathematical Statistics*, pp. 326–339, 1948.

[48] A. Wald, "Sequential tests of statistical hypotheses," in *Breakthroughs in Statistics*. Springer, 1992, pp. 256–298.

[49] V. Aalo and R. Viswanathan, "Asymptotic performance of a distributed detection system in correlated gaussian noise," *IEEE Transactions on Signal Processing*, vol. 40, no. 1, pp. 211–213, 1992.

[50] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005, DySPAN'05*, Baltimore, Maryland, USA, November 2005, pp. 131–136.

[51] S. Atapattu, C. Tellambura, and H. Jiang, "Energy detection based cooperative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 4, pp. 1232–1241, 2011.

[52] D. Teguig, B. Scheers, and V. Le Nir, "Data fusion schemes for cooperative spectrum sensing in cognitive radio networks," in *IEEE Communications and Information Systems Conference (MCC), 2012 Military*, Gdansk, Poland, October 2012, pp. 1–7.

[53] Z. Li, P. Shi, W. Chen, and Y. Yan, "Square-law combining double-threshold energy detection in nakagami channel." *International Journal of Digital Content Technology & its Applications*, vol. 5, no. 12, 2011.

[54] H. Sun, "Collaborative spectrum sensing in cognitive radio networks," Ph.D. dissertation, The University of Edinburgh, 2011.

[55] M. K. Simon and M.-S. Alouini, *Digital communication over fading channels*. John Wiley & Sons, 2005, vol. 95.

[56] R. S. Blum, S. A. Kassam, and H. V. Poor, "Distributed detection with multiple sensors i. advanced topics," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 64–79, 1997.

[57] T. Weiss, "A diversity approach for the detection of idle spectral resources in spectrum pooling systems," in *Proceedings of 48th International Scientific Colloquium*, Ilmenau, Germany, September 2003, pp. 37–38.

[58] J. Ma, G. Zhao, and Y. Li, "Soft combination and detection for cooperative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4502–4507, 2008.

[59] E. Visotsky, S. Kuffner, and R. Peterson, "On collaborative detection of tv transmissions in support of dynamic spectrum sharing," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005, DySPAN'05.*, Baltimore, Maryland, USA, November 2005, pp. 338–345.

[60] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[61] S. H. Javadi, "Detection over sensor networks: a tutorial," *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, no. 3, pp. 2–18, March 2016.

[62] Z. Chair and P. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, no. 1, pp. 98–101, Jan 1986.

[63] S. C. Thomopoulos, R. Viswanathan, and D. C. Bougoulias, "Optimal decision fusion in multiple sensor systems," *IEEE Transactions on Aerospace and Electronic Systems*, no. 5, pp. 644–653, 1987.

[64] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University press, 2007.

[65] M. J. Osborne, *An introduction to game theory*. Oxford University Press New York, 2004, vol. 3, no. 3.

[66] J. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[67] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994.

[68] J. V. Neumann, "Zur theorie der gesellschaftsspiele," *Mathematische Annalen*, vol. 100, pp. 295–320, 1928. [Online]. Available: http://eudml.org/doc/159291

[69] V. Chvatal, *Linear programming*. Macmillan, 1983.

[70] A. Charnes and W. W. Cooper, "Management models and industrial applications of linear programming," *Management Science*, vol. 4, no. 1, pp. 38–91, 1957.

[71] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 195–259, 2009.

[72] R. D. McKelvey and T. R. Palfrey, "An experimental study of the centipede game," *Econometrica: Journal of the Econometric Society*, pp. 803–836, 1992.

[73] Y. C. Chen, N. Van Long, and X. Luo, "Iterated strict dominance in general games," *Games and Economic Behavior*, vol. 61, no. 2, pp. 299–315, November 2007.

[74] D. Bernheim, "Rationalizable strategic behavior," *Econometrica*, vol. 52, pp. 1007–1028, 1984.

[75] P. Weirich, *Equilibrium and rationality: game theory revised by decision rules.* Cambridge University Press, 2007.

[76] H. Chan and A. Perrig, "Security and privacy in sensor networks," *Computer*, vol. 36, no. 10, pp. 103–105, 2003.

[77] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[78] M. Barni, M. Fontani, and B. Tondi, "A universal technique to hide traces of histogram-based image manipulations," in *Proceedings of the ACM Workshop on Multimedia and Security*, New York, NY, USA, September 2012, pp. 97–104. [Online]. Available: http://doi.acm.org/10.1145/2361407.2361424

[79] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 428–445, 2013.

[80] Y. Yang, Y. L. Sun, S. Kay, and Q. Yang, "Defending online reputation systems against collaborative unfair raters through signal modeling and trust," in *Proceedings of the 24th ACM Symposium on Applied Computing*, Honolulu, Hawai, USA, 9-12, March, 2009.

[81] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Chicago, IL, USA, May 2005, pp. 46–57.

[82] R. Chen, J.-M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 25–37, 2008.

[83] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[84] A. M. Wyglinski, M. Nekovee, and T. Hou, *Cognitive radio communications and networks: principles and practice.* Academic Press, 2009.

[85] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[86] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication*, vol. 2011, no. 4, pp. 40–62, 2011.

[87] T. C. Clancy and N. Goergen, "Security in cognitive radio networks: Threats and mitigation," in *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008, CrownCom'08*, Singapore, May 2008, pp. 1–8.

[88] Z. Chen, T. Cooklev, C. Chen, and C. Pomalaza-Ráez, "Modeling primary user emulation attacks and defenses in cognitive radio networks," in *IEEE 28th International Performance Computing and Communications Conference, 2009*, Phoenix, Arizona, USA, December 2009, pp. 208–215.

[89] Z. Jin, S. Anand, and K. Subbalakshmi, "Mitigating primary user emulation attacks in dynamic spectrum access networks using hypothesis testing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 74–85, 2009.

[90] H. Sharma and K. Kumar, "Primary user emulation attack analysis on cognitive radio," *Indian Journal of Science and Technology*, vol. 9, no. 14, 2016.

[91] A. W. Min, K. G. Shin, and X. Hu, "Attack-tolerant distributed sensing for dynamic spectrum access networks," in *17th IEEE International Conference on Network Protocols, 2009, ICNP'09*, Princeton, New Jersey, USA, October 2009, pp. 294–303.

[92] T. Qin, H. Yu, C. Leung, Z. Shen, and C. Miao, "Towards a trust aware cognitive radio architecture," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 86–95, 2009.

[93] P. Kaligineedi, M. Khabbazian, and V. K. Bhargava, "Secure cooperative sensing techniques for cognitive radio systems," in *IEEE International Conference on Communications, 2008*, Beijing, China, May 2008, pp. 3406–3410.

[94] A. Attar, H. Tang, A. V. Vasilakos, F. R. Yu, and V. C. Leung, "A survey of security challenges in cognitive radio networks: Solutions and future research directions," *Proceedings of the IEEE*, vol. 100, no. 12, pp. 3172–3186, 2012.

[95] W. Wang, H. Li, Y. Sun, and Z. Han, "Attack-proof collaborative spectrum sensing in cognitive radio networks," in *43rd IEEE Annual Conference on Information Sciences and Systems, 2009, CISS'09*, Baltimore, MD, USA, March 2009, pp. 130–134.

[96] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *2006 IEEE International Conference on Communications*, vol. 4, Istanbul, Turkey, June 2006, pp. 1658–1663.

[97] S. Sodagari, A. Attar, V. C. Leung, and S. G. Bilén, "Denial of service attacks in cognitive radio networks through channel eviction triggering," in *IEEE Global Telecommunications Conference, 2010, GLOBECOM'10*, Miami, Florida, USA, December 2010, pp. 1–5.

[98] J. J. Fitton, "Security considerations for software defined radios," in *Proceeding of the 2002 Software Defined Radio Technical Conference*, vol. 1, 2002.

[99] C. Li, A. Raghunathan, and N. K. Jha, "An architecture for secure software defined radio," in *Proceedings of the European Design and Automation Association Conference on Design, Automation and Test in Europe*, Lausanne, Switzerland, April 2009, pp. 448–453.

[100] S. Xiao, J. M. Park, and Y. Ye, "Tamper resistance for software defined radio software," in *2009 33rd Annual IEEE International Computer Software and Applications Conference*, vol. 1, Seattle, Washington, USA, July 2009, pp. 383–391.

[101] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[102] E. H. Dinan and B. Jabbari, "Spreading codes for direct sequence cdma and wideband cdma cellular networks," *IEEE Communications Magazine*, vol. 36, no. 9, pp. 48–54, 1998.

[103] , "Spectrum policy task force report et docket no. 02-135," *US Federal Communications Commission*, 2002.

[104] Z. Jin, S. Anand, and K. Subbalakshmi, "Detecting primary user emulation attacks in dynamic spectrum access networks," in *2009 IEEE International Conference on Communications*, Dresden, Germany, June 2009, pp. 1–5.

[105] K. Dogancay and D. A. Gray, "Closed-form estimators for multi-pulse tdoa localization," in *Proceedings of the Eighth IEEE International Symposium on Signal Processing and Its Applications, 2005*, vol. 2, Sydney, NSW, Australia, August 2005, pp. 543–546.

[106] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *IEEE Global Telecommunications Conference, 2001, GLOBECOM'01*, vol. 5, San Antonio, Texas, USA, November 2001, pp. 2926–2931.

[107] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *2010 IEEE Symposium on Security and Privacy*, Oakland, California, USA, May 2010, pp. 286–301.

[108] C. N. Mathur and K. P. Subbalakshmi, "Digital signatures for centralized dsa networks," in *2007 4th IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, USA, January 2007, pp. 1037–1041.

[109] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of byzantine attacks," *IEEE Transactions on Signal Processing,*, vol. 57, no. 1, pp. 16–29, 2009.

[110] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of byzantine attack in large wireless sensor networks," in *IEEE Military Communications Conference, 2006, MILCOM'06*, Washington, DC, USA, October 2006, pp. 1–4.

[111] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley Interscience, 1991.

[112] A. S. Rawat, P. Anand, H. Chen, and P. K. Varshney, "Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 774–786, February 2011.

[113] A. Vempaty, K. Agrawal, P. Varshney, and H. Chen, "Adaptive learning of byzantines' behavior in cooperative spectrum sensing," in *Proceedings of WCNC'11, IEEE Conference on Wireless Communications and Networking*, Cancun, Mexico, March 2011, pp. 1310–1315.

[114] A. Rawat, P. Anand, H. Chen, and P. Varshney, "Countering byzantine attacks in cognitive radio networks," in *Proceedings of ICASSP'10, IEEE International Conference on Acoustics Speech and Signal Processing*, Dallas, Texas, USA, March 2010, pp. 3098–3101.

[115] E. Noon and H. Li, "Defending against hit-and-run attackers in collaborative spectrum sensing of cognitive radio networks: A point system," in *IEEE 71st Vehicular Technology Conference (VTC 2010-Spring), 2010*, Taipei, Taiwan, May 2010, pp. 1–5.

[116] H. Li and Z. Han, "Catching attacker(s); for collaborative spectrum sensing in cognitive radio systems: An abnormality detection approach," in *IEEE Symposium on New Frontiers in Dynamic Spectrum, 2010, DySPAN'10*, Singapore, Singapore, April 2010, pp. 1–12.

[117] B. Kailkhura, S. Brahma, Y. S. Han, and P. K. Varshney, "Optimal distributed detection in the presence of byzantines," in *ICASSP 2013, IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 27-31, May 2013, pp. 2925–2929.

[118] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.

[119] http://it.mathworks.com/help/optim/.

[120] L. S. Shapley, "A note on the lemke-howson algorithm," in *Pivoting and Extension*. Springer, 1974, pp. 175–189.

[121] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 325–343, Mar 2000.

[122] P. Pakzad and V. Anantharam, "A new look at the generalized distributive law," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1132–1155, June 2004.

[123] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, January 1962.

[124] S. Verdu and H. V. Poor, "Abstract dynamic programming models under commutativity conditions," *SIAM Journal on Control and Optimization*, vol. 25, no. 4, pp. 990–1006, 1987.

[125] J. Pearl and M. Kaufmann, "Probabilistic reasoning in intelligent systems, san mateo, ca," *Cal.: Morgan Kaufmann*, 1988.

[126] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, Jan 1986.

[127] K. W. Choi and E. Hossain, "Estimation of primary user parameters in cognitive radio systems via hidden markov model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 782–795, Feb 2013.

[128] I. A. Akbar and W. H. Tranter, "Dynamic spectrum allocation in cognitive radio using hidden markov models: Poisson distributed case," in *IEEE Proceedings of SoutheastCon*, Richmond, Virginia, USA, March 2007, pp. 196–201.

[129] T. Jiang, H. Wang, and A. V. Vasilakos, "Qoe-driven channel allocation schemes for multimedia transmission of priority-based secondary users over cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 7, pp. 1215–1224, August 2012.

[130] D. J. MacKay, *Information theory, inference and learning algorithms.* Cambridge university press, 2003.

[131] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[132] T. Richardson and R. Urbanke, *Modern coding theory.* Cambridge University Press, 2008.

[133] Y. Mao and A. H. Banihashemi, "A heuristic search for good low-density parity-check codes at short block lengths," in *IEEE International Conference on Communications, 2001, ICC'01*, vol. 1, Helsinki, Finland, June 2001, pp. 41–44.

[134] B. Kailkhura, S. Brahma, and P. Varshney, "On the performance analysis of data fusion schemes with byzantines," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 7411–7415.

Index