2015

Quantifying Trust and Reputation for Defense against Adversaries in Multi-Channel Dynamic Spectrum Access Networks

Shameek Bhattacharjee
University of Central Florida

Quantifying Trust and Reputation for Defense against Adversaries in Multi Channel Dynamic Spectrum Access Networks

by

Shameek Bhattacharjee
B.Tech., West Bengal University of Technology, 2009, India
M.S., University of Central Florida, 2011, USA

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2015

Major Professor: Mainak Chatterjee

ABSTRACT

Dynamic spectrum access enabled by cognitive radio networks are envisioned to drive the next generation wireless networks that can increase spectrum utility by opportunistically accessing unused spectrum. Due to the policy constraint that there could be no interference to the primary (licensed) users, secondary cognitive radios have to continuously sense for primary transmissions. Typically, sensing reports from multiple cognitive radios are fused as stand-alone observations are prone to errors due to wireless channel characteristics. Such dependence on cooperative spectrum sensing is vulnerable to attacks such as Secondary Spectrum Data Falsification (SSDF) attacks when multiple malicious or selfish radios falsify the spectrum reports. Hence, there is a need to quantify the trustworthiness of radios that share spectrum sensing reports and devise malicious node identification and robust fusion schemes that would lead to correct inference about spectrum usage.

In this work, we propose an anomaly monitoring technique that can effectively capture anomalies in the spectrum sensing reports shared by individual cognitive radios during cooperative spectrum sensing in a multi-channel distributed network. Such anomalies are used as evidence to compute the trustworthiness of a radio by its neighbors. The proposed anomaly monitoring technique works for any density of malicious nodes and for any physical environment. We propose an optimistic trust heuristic for a system with a normal risk atti-

tude and show that it can be approximated as a beta distribution. For a more conservative system, we propose a multinomial Dirichlet distribution based conservative trust framework, where Josang's Belief model is used to resolve any uncertainty in information that might arise during anomaly monitoring. Using a machine learning approach, we identify malicious nodes with a high degree of certainty regardless of their aggressiveness and variations introduced by the pathloss environment. We also propose extensions to the anomaly monitoring technique that facilitate learning about strategies employed by malicious nodes and also utilize the misleading information they provide.

We also devise strategies to defend against a collaborative SSDF attack that is launched by a coalition of selfish nodes. Since, defense against such collaborative attacks is difficult with popularly used voting based inference models or node centric isolation techniques, we propose a channel centric Bayesian inference approach that indicates how much the collective decision on a channels occupancy inference can be trusted. Based on the measured observations over time, we estimate the parameters of the hypothesis of anomalous and non-anomalous events using a multinomial Bayesian based inference. We quantitatively define the trustworthiness of a channel inference as the difference between the posterior beliefs associated with anomalous and non-anomalous events. The posterior beliefs are updated based on a weighted average of the prior information on the belief itself and the recently observed data.

Subsequently, we propose robust fusion models which utilize the trusts of the nodes to improve the accuracy of the cooperative spectrum sensing decisions. In particular, we

propose three fusion models: (i) optimistic trust based fusion, (ii) conservative trust based fusion, and (iii) inversion based fusion. The former two approaches exclude untrustworthy sensing reports for fusion, while the last approach utilizes misleading information. All schemes are analyzed under various attack strategies. We propose an asymmetric weighted moving average based trust management scheme that quickly identifies on-off SSDF attacks and prevents quick trust redemption when such nodes revert back to temporal honest behavior. We also provide insights on what attack strategies are more effective from the adversaries' perspective.

Through extensive simulation experiments we show that the trust models are effective in identifying malicious nodes with a high degree of certainty under variety of network and radio conditions. We show high true negative detection rates even when multiple malicious nodes launch collaborative attacks which is an improvement over existing voting based exclusion and entropy divergence techniques. We also show that we are able to improve the accuracy of fusion decisions compared to other popular fusion techniques. Trust based fusion schemes show worst case decision error rates of 5% while inversion based fusion show 4% as opposed majority voting schemes that have 18% error rate. We also show that the proposed channel centric Bayesian inference based trust model is able to distinguish between attacked and non-attacked channels for both static and dynamic collaborative attacks. We are also able to show that attacked channels have significantly lower trust values than channels that are not– a metric that can be used by nodes to rank the quality of inference on channels.

To my parents and teachers.

ACKNOWLEDGMENTS

I am grateful to my advisor Dr. Mainak Chatterjee for guiding, supporting and believing in me over the years of my Ph.D. Without his experienced mentoring, my dream of earning a Ph.D would not have come true. I would like to express sincere appreciation to my committee members Dr. Necati F. Catbas, Dr. Ratan Guha, Dr. Damla Turgut and Dr. Cliff Zou for serving in my committee. Their constructive feedback and comments have helped me in improving my dissertation. I would also like to thank Dr. Kevin Kwiat and Dr. Charles Kamhoua of the Air Force Research Lab, Rome, NY for providing me with collaborative research opportunities. I would like to thank all my colleagues at the NetMoC laboratory and all my friends who have always inspired me while working as a Ph.D. student. In addition, I would like to thank the Air Force Research Lab (AFRL) and Department of Electrical Engineering and Computer Science at the University of Central Florida for partially funding my studies. I would like to acknowledge the role of my parents and family members for motivating and supporting my desire of pursuing higher studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

In this chapter, we will discuss some background on the concepts of cognitive radio networks, vulnerabilities associated with protocols that lead to threats, the salient contributions of this dissertation along with the benefits and finally the organization the dissertation.

1.1 Cognitive Radio Networks

Radio spectrum allocation and management have traditionally followed a 'command-and-control' approach. Regulators like the Federal Communications Commission (FCC) in the United States allocate spectrum to specific services under restrictive licenses. The restrictions specify the technologies to be used and the services to be provided, thereby constraining the ability to make use of new technologies and the ability to redistribute the spectrum to higher valued services. These limitations have motivated a paradigm shift from static spectrum allocation towards a more 'liberalized' notion of dynamic spectrum management in which secondary networks/users (non-license holders) can 'borrow' idle spectrum from those who hold licensees (i.e., primary networks/users), without causing harmful interference to the latter– a notion commonly referred to as dynamic spectrum access (DSA) or open spectrum access [1]. It is envisioned that DSA networks enabled with cognitive radio

(CR) devices [2, 3] will bring about radical changes in wireless communications that would opportunistically exploit unused spectrum bands. However, the *open* philosophy of the unmanaged/unlicensed spectrum makes the cognitive radio networks susceptible to events that prevent them from communicating effectively. Just like traditional radios, cognitive radios are not only susceptible to interference but also need spectrum assurance. Unlike traditional radios, cognitive radios constantly monitor the spectrum and intelligently share the spectrum in an opportunistic manner, both in licensed and unlicensed bands. The most important regulatory aspect of these networks is that unlicensed cognitive radios must relinquish their operating channels and move to another available channel as soon as they learn or sense the presence of a licensed user on that channel [4].

As spectrum is made available to unlicensed users, it is expected that all such users will follow the regulatory aspects and adhere to the spectrum sharing and access rules. However, the inherent design of cognitive radios exposes its configuration options to the controlling entity in an effort to make the operational parameters flexible and tunable. As a consequence, the reconfigurability and adaptability features open up avenues for manipulation as well. Moreover, problems arise when regulatory constraints are not followed. Also, learning features of the cognitive radios can be manipulated. A radio can be induced to learn false information by malicious or selfish entities, the effect of which can sometimes propagate to the entire network. It is apparent that the inherent design, flexibility and openness of opportunistic spectrum usage have opened avenues of attacks and made cogni-

tive radio networks susceptible to various genres of vulnerabilities including non-compliance of regulations.

The vacancy or occupancy of primary user's signal is known as *channel occupancy status*. This can be assessed through *local spectrum sensing* by each CR node. However, due to typical wireless channel impairments like signal fading, multipath shadowing, a stand-alone radio's local sensing cannot always infer the true occupancy status of a channel. Hence radios participate in cooperative spectrum sensing [5, 6], where an inference on the occupancy status of a channel is made after fusing multiple local sensing results advertised by various CR nodes.

1.2 Vulnerabilities In Cooperative Spectrum Sensing

Cooperative spectrum sensing can be vulnerable when some malicious nodes share *false* local sensing reports to others. In such cases, the fused decision may be altered, hence jeopardizing the reliability of cooperative spectrum sensing. Such phenomenon where local sensing result is manipulated is known as Spectrum Sensing Data Falsification (SSDF) [7, 8] or Byzantine attack. A malicious radio can advertise 'occupied' as ' available' inducing a policy violation or advertise 'available' as 'occupied' causing denial of spectrum usage. In adversarial, military, and heterogeneous competitive networks such actions are not surprising where an adversary wants to cripple the operation of others in the network. Apart from this, there are also cases where a node's permanent spatial orientation is such that its reports are

not suitable for use by other nodes. The adversary may vary its attack strategies based on different objectives. Hence there is a need to evaluate the trustworthiness of radios before considering their local spectrum sensing reports. A trust aware selection of cooperative cognitive radios can filter out spurious information and preserve the correctness of spectrum occupancy inference. Moreover, identification or isolation of such malicious nodes are also required for policy enforcement and liability assignment. Appropriate criterion needs to be designed for identification and banning of malicious nodes.

1.3 Contributions of this Work

In this work, we provide an anomaly monitoring technique for distributed DSA networks where binary spectrum sensing results on multiple channels are shared [9]. The anomaly monitoring technique gathers evidence that is able to indicate cooperative or dishonest behavior of CR nodes. The anomaly monitoring technique is able to capture anomalies of malicious nodes which employ various attack strategies where nodes might also collaborate to launch such attacks. The monitoring technique is based on predicting the bounds on the received power using received signal strength lateration and spatio-spectral geometry of CR nodes and primary transmitters, while preserving location privacy requirement of CR nodes. We provide an analytical model of improving the monitored evidence using long-term information on channel occupancy statistics [10]. We propose *Observed Invert Sequence* as

an extension that facilitates important functions like learning the attack pattern of malicious nodes and propose a way to utilize misleading information provided by malicious nodes.

Based on the evidence gathered from the anomaly monitoring technique, we propose different trust models that differentiates between honest and dishonest nodes with a high degree of certainty and preserve accuracy/integrity of cooperative spectrum sensing. First, we propose an *optimistic trust heuristic* inspired from an approximation of a Beta distribution model [9] that assigns a trust value between 0 and 1 to all neighboring CR nodes. We also provide a second measure that quantifies *confidence* on the computed trust value given imperfect monitoring or incomplete information. The combination of optimistic trust heuristic and confidence is used to rank nodes in terms of the quality of spectrum sensing data they provide. Subsequently, we propose an *optimistic trust based fusion* scheme using the optimistic trust heuristic as a criterion to filter out sensing reports from potentially untrustworthy nodes [9]. The proposed trust based fusion works better than blind fusion for both collaborative and non-collaborative SSDF attacks.

Using the optimistic trust heuristic, we provide a log-weighted trust metric that can effectively distinguish between malicious and honest nodes. Using a combination of the log weighted trust metric and observed invert sequence, we propose an intelligent inversion based fusion schema [11] where instead of excluding sensing reports of malicious nodes, we utilize their misleading information to the advantage of the network. We observe significant improvement in cooperative sensing performance from blindly fused spectrum data

or exclusionary approaches that disregard information from malicious nodes or isolate such nodes.

We propose various types of SSDF strategies on multi-channel systems and quantify two types of attack measures. We investigate how they affect the utility of CR network employing majority voting for fusion. In particular, we show that trust values do indeed reflect how aggressive a malicious node is.

We provide a conservative trust model for a more conservative system with a higher risk attitude using logic of uncertain probabilities (subjective logic), and Dirichlet distribution. We discuss a machine learning approach that uses conservative trust values to effectively identify and distinguish between malicious and honest nodes regardless of type of pathloss environment, how aggressive a malicious node is, or the density of malicious nodes in the network, all of which affect the monitoring outcome. We also propose a conservative trust based fusion similar to the optimistic trust based fusion for disregarding untrustworthy reports. Additionally, nodes in network may engage in on-off attacks where a malicious node may initially build a high reputation by behaving honestly and then behaving maliciously later. Such sudden changes in behavior cannot be captured quick enough using traditional weighted moving averages. As a solution to this, we propose an asymmetric weighted moving average scheme for quickly identifying on-off attacks.

Finally, we provide a Bayesian inference based trust framework that indicates how much the collective decision on a channel's occupancy can be trusted, in presence of selfish nodes who collaboratively modify occupancy reports on preferred channels. Then we show

that the framework is generic enough to be extended to any cooperative decision making system.

1.4 Benefits of this Work

Our work provides key insights into multi-channel attack strategies and salient concepts on trust and reputation scoring in the context of SSDF attacks in DSA networks.

Our framework gives accurate results under collaborative SSDF attacks even when the fraction of malicious nodes is high which is a significant improvement from voting based or entropy divergence techniques. Our framework also takes care of the difficulties faced by voting based mechanisms when all participating nodes may not be in the primary transmitter's coverage area. The proposed schemes are applicable on different pathloss environments.

The proposed reputation based framework does not need location information of CR nodes thereby preserving location privacy and obviating the possibility of location falsification at the same time. Location privacy is even more important in a CR network to guard against various RF manipulations, primary user emulation attacks, and belief manipulation.

Our channel centric Bayesian framework that quantifies trustworthiness of spectrum decisions is very generic and can be extended to any cooperative decision making system for quantifying decision reliability under imperfect monitoring and adversaries.

1.5 Organization of the Dissertation

The dissertation has been organized as follows. Chapter 2 discusses the preliminary concepts related to DSA networks and their associated vulnerability and security issues. Chapter 3 discusses the preliminary background on trust metrics, reputation systems, SSDF attacks, and some open issues on trust and reputation scoring in secondary DSA network. Chapter 4, discusses our anomaly monitoring technique using bound prediction on received power to gather evidence that indicate presence of anomalous reports. We also discuss special monitoring extensions to the technique which is later used for exploiting the misleading information. Chapter 5 discusses the optimistic trust heuristic for a system with normal risk attitude and the disadvantages of using such a model for a conservative system. We also propose a conservative trust model that can also include effect of uncertainty in collected evidence. Chapter 6 discusses how trust computation from each of the models are applied for malicious node identification, robust information fusion and resilient reputation management of nodes. Chapter 7 discusses a channel centric Bayesian inference framework for defending against collaborative selfish nodes who launch SSDF attacks with channel preference. Chapter 8 gives a comprehensive discussion on the results obtained via simulations. Chapter 9 concludes this dissertation.

CHAPTER 2: CRNs AND VULNERABILITIES

In this chapter, we discuss architectural aspects and key operational weaknesses that introduce avenues for manipulation by adversaries. Then we categorize and classify different types of vulnerabilities and security threats in DSA networks. We discuss the threat models and various possible attack strategies that are relevant to this dissertation.

2.1 Architectural Aspects and Operational Weaknesses

Before we discuss the vulnerabilities of DSA networks, let us first present the architectural aspects of cognitive radios. In particular, we focus on the cognitive functionalities and the architectural aspects of the network that make them prone to different genres of attack.

A typical cognitive radio consists of a sensor, a radio, a knowledge database, a learning engine, and a reasoning engine. A cognitive radio continuously learns from its surroundings and adapts its operational parameters to the statistical variations of incoming radio frequency (RF) stimulus [2]. The objective is to select a set of parameters based on knowledge, experience, cognition, and policies, in such a way so as to produce outputs that optimize some objective function. An architectural overview of the cognitive engine, input parameters, observable metrics, policies and objective functions is shown in Fig. 2.1. In the cognitive

domain, knowledge or cognizance is obtained from awareness of surroundings, based on input statistics from sensory observations and other network parameters.

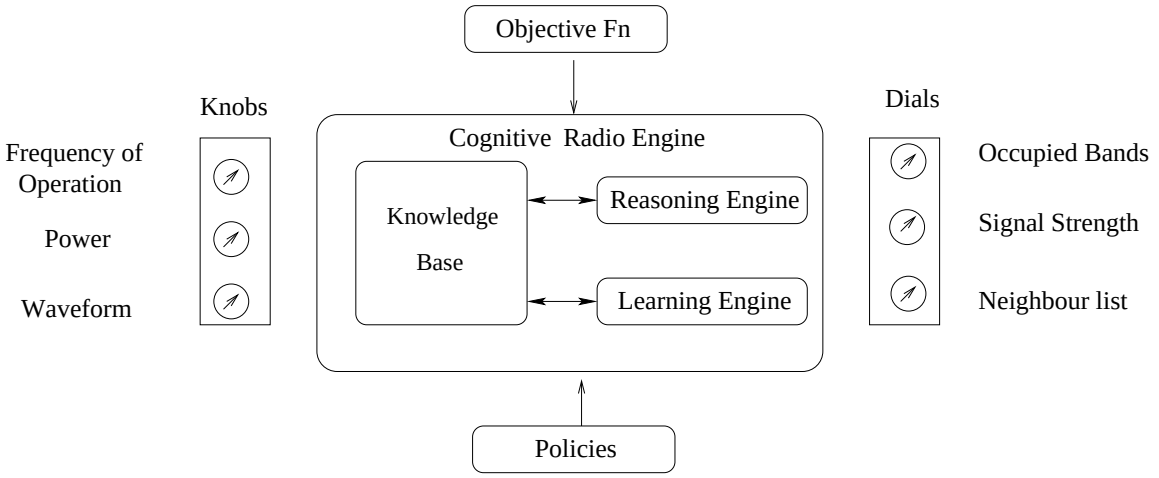


Figure 2.1: Architectural overview of cognitive radio

Cognitive radios usually have a programming interface that exposes the configuration options to a controlling entity. The controlling entity could be the service provider that deploys the cognitive radios (base station, access point, etc.) who needs to frequently change the operational parameters– for example, the operating band, access policies, transmission power, and modulation schemes [12, 13]. As it is rather impractical to have physical connections with the cognitive radios, the programming of the radios is usually done over-the-air. In the absence of an infrastructure, there might not be any controlling entity and therefore the programming capability could be limited.

2.2 Main Operational Aspects

Local Spectrum Sensing and Local Decision: Cognitive radios have to decide on the vacancy of channels before using them [2, 3, 12]. So the radios scan for presence or absence of primary transmission over multiple channels. The easiest and most cost-effective form of sensing the primary's presence is energy detection [14]. This process is termed as *local sensing* as it is done by a stand-alone cognitive radio.

The local sensing result on a channel may be represented as raw energy values. A radio deciding on the vacancy compares this energy detected on a channel with a *threshold*; if energy is greater than the threshold, the channel is inferred to be 'occupied' (represented as 1) by a primary or a secondary; else it is inferred as 'empty' (represented as 0). Hence for a cognitive radio operating on a multi-channel spectrum, the local sensing result is either a binary vector of 1's and 0's known as *hard local decisions*.

Collaborative or Cooperative Sensing: In collaborative and cooperative spectrum sensing, radios share their sensed information with others; hence the level of cooperation has a direct effect on the efficiency of resource usage. This is because all radios are exposed to typical wireless characteristics like signal fading and noise which may result in wrong inference [5]. To reduce the level of uncertainty, cognitive radios often employ cooperative spectrum sensing, [6], [15], [16], [17], [18], where the spectrum decision is based on a fusion of opinions provided by a number of radios in the network. The fused spectrum decision is known as *global decision*. The common assumption for sending local sensing results is over a common

control channel. Sharing hard local decisions have also been in focus due to less cost and bandwidth requirements.

- In an infrastructured CR network, the local sensing results are sent to the central fusion center (FC) which combines the local results in accordance with a suitable fusion technique. If hard decisions are shared, the popular fusion techniques are plurality voting or majority voting etc. If soft decisions are shared, the FC compares each value with the common network threshold to convert them into a vector of 1's and 0's. Then it applies one of the fusion techniques. The fusion center decides on the vacancy and allocation of channels.

- In contrast, in the distributed ad hoc mode, the local sensing results are advertised to all neighbors within a certain sharing radius. A radio fuses the local sensing results of it's neighbors. The process of fusing data from other radios usually entails cooperation, and thus collaborative or cooperative sensing is usually employed. Usually a common control channel and a transmit power is agreed upon for advertising local sensing results.

2.3 Categories of Vulnerabilities in Cooperative Spectrum Sensing

There is always a difference (both temporal and spatial) between the collected data and the result of the fusion. Such dependence on information from other radios makes the collaboration vulnerable to malicious radios which could provide misleading data. Moreover,

such spectrum usage sharing might indirectly reveal the location information of a radio violating its location privacy rights. However, measures on preserving the location privacy in cooperative spectrum sensing have been proposed in [19].

We classify the various categories of vulnerabilities into three categories; objective of attackers, impact on victims, and nature of manipulation.

2.3.1 Objective of adversarial attackers

The objectives of an attacker have a direct correlation with the way the attacks are launched, and therefore they determine the nature of attacks. Hence we classify dishonest nodes or dishonest behavior into the following two types:

- Selfish Attacks: The attacker's motive is to acquire more spectrum for its own use by preventing others from competing for the channels and unfairly occupying their share. In this type of attack, adversaries will defy the protocols and policies only if they are able to benefit from them.

- Malicious Attacks: The attacker's only objective is to create hindrance for others and does not necessarily aim at maximizing own benefits. They do not have any rational objective and defy protocols and policies to induce losses to others.

2.3.2 Impact of attack on the victims

- Direct Attack: In direct attacks, the objective of the adversary is denial or refusal of communication or service whenever possible. An example would be to somehow make the radio believe that primary incumbent is present, when in-fact the primary is not present. This is a classical example of denial of service attack where honest cognitive nodes are denied authorized access. Another example is jamming them by sending interfering signals on a channel agreed upon by a transmitter-receiver pair for data communication. We discuss several subclasses of such attacks in the Section 2.3.3.1.

- Induced Attack: In induced attacks, the attacks are related to policy violation and breach of regulation. There is usually a significant delay between the actual execution of the attack and its effect on the victim. It often has serious legal consequences as the effects are associated with breach of regulations and agreements. For example, inducing unauthorized spectrum access through a policy violation by making a radio believe that the primary is not present when in-fact the primary is present, thus causing a regulatory violation.

2.3.3 Nature of manipulation

Based on the nature of manipulation there can be the following categories of threats:

2.3.3.1 Sensory manipulation

As obvious from the term, the attack is done in such a way that sensors those sense the presence of primaries are provided with misleading information. Spoofing faulty sensor information will cause the radios to make incorrect decisions about spectral occupancy and may select configurations or set of parameters that provides sub-optimal performance. Primary user emulation attacks is an example of sensory manipulation where the sensors perceive a spoofed signal that resemble the signal of a licensed user and is led to believe that spectrum is not available for use. This type of attack can be quickly launched and therefore is a type of immediate denial attack.

1. Direct sensory manipulation: Malicious nodes may alter sensory input statistics in such a manner so as to deny communication opportunities to others. For example, a malicious node can simply emit spurious signals with signal properties similar to that of a primary incumbent thereby impersonating the presence of the primary incumbent. Thus, a sensor would fail to detect the spectrum vacancy even when the primary is not transmitting.

2. Induced sensory manipulation: Here, the sensory input is altered to make a sensor fail to identify the presence of the primary. This can be done by a variety of ways like raising the noise floor, masking signals, and advertising lower signal to noise ratio values during cooperative sensing. All these will make a radio believe that the primary is not present and will be tempted to use the channel which will induce interference to

the primary. While the effect of interference is immediate, a radio may be banned after repeated occurrences of such induced interference. Thus, there is a time lag between the time of execution of the attack and its effect to take place.

2.3.3.2 Belief manipulation

This type of attack can be aimed at procedural and ontological cognitive radios that use learning and experience. The radios learn to associate the temporal and spatial characteristics of the channel occupancy that are faulty. Another example would be that an attacker can introduce a jamming signal whenever a cognitive radio device switches to higher modulation rates, thus forcing it to operate on lower modulation rate. It is led to believe that switching to higher modulation rate causes interference and it employs lower data rates, and may never try higher data rates, given the past experience.

1. Direct belief manipulation: This attack is closely related to cooperative spectrum sensing, where multiple radios may lie about their opinion on spectral occupancy. If such modified opinions are shared, the fusion outcome is wrong. Obviously the severity of such manipulation depends on how a node fuses the information. The secondary spectrum data falsification attack is an example of a direct belief manipulation in which spurious occupancy information is sent to honest radios.

2. Induced belief manipulation: Here the learning radios associate wrong temporal and spatial characteristics of the RF environment and orient their functionalities and con-

figurations to an operating state that results in a sub-optimal performance. As radios employ learning algorithms, case-driven memory and case-based learning, spurious inputs pollute the inference and knowledge base significantly. So when the learning stage is affected, the decision phase is also affected. For example, few dynamic spectrum access algorithms gather channel access statistics for PUs in an attempt to predict when the channel will be idle [20]. If attackers keep spoofing modified occupancy information on a channel, it will affect the long term behavior of the radio.

2.4 Secondary Spectrum Data Falsification (SSDF) or Byzantine attacks

A Byzantine failure or an SSDF attack in secondary networks [8], [21], occurs when dishonest attackers lie or modify secondary spectrum sensing data, which may cripple the fusion center or individual radios who are unable to correctly determine the status of primary's presence. This attack exploits the cooperative nature of spectrum sensing where an attacker sends false spectrum data to the fusion center or data collector, thus inducing erroneous decisions on spectral usage. There are three ways in which a Byzantine attack can be launched.

1. Denial SSDF: The adversary may advertise 0 (not occupied) as 1 (occupied) thus causing the fusion/channel allocation center to believe that primary is present, thus restricting channel access. This attack comes under both short term and denial attack,

as interpreting empty spectrum as occupied means that a radio cannot use the spectrum with immediate effect.

2. Induce SSDF: The adversary may advertise 1 as 0 thus causing harmful interference to primary incumbent. Repeated occurrence of such breach of policies may cause the radio to be barred temporarily or banned permanently from the network. Since repeated occurrence of this instance is necessary, it is a long term or induce attack. This is distinct from the previous case which was a denial attack and is achieved quickly.

2.5 SSDF Attack Strategies

Attack strategies are dictated by many factors and these factors are often interdependent. They include the following:

- The extent of aggression of a dishonest node, or the amount of damage it wants to inflict.
- The goals or objectives it seeks to achieve through dishonest behavior.
- Whether multiple dishonest nodes collude or not while modifying local spectrum sensing reports.

2.5.1 Magnitude of attack

The magnitude of attack defines the level of aggression of a particular dishonest node. The value of magnitude of attack represents the short term or long term fraction of channels the dishonest nodes falsifies upon. It may be noted that very low magnitude of attack may not affect the network's operations significantly, while very high magnitudes of attack may facilitate easy detection. More details on magnitude of attack are given in Chapter 4.

2.5.2 Collaborative vs. non-collaborative strategies

1. Non-Collaborative SSDF: The dishonest nodes *do not agree* upon the channels they choose to attack. Each malicious node launches independent attack.

2. Collaborative SSDF: The dishonest nodes agree upon the subset of channels they attack. It increases the chances of circumventing the fusion rule and may be more effective form of attack under certain conditions. However, collaboration among malicious nodes increases cost of attack and requires synchronization. Usually the nodes agree upon the set of channels to attack.

2.5.3 Malicious vs. selfish nodes

1. Malicious SSDF nodes: This strategy is employed, when the objective of malicious nodes is to inflict maximum possible damage to the other nodes. These nodes are not seeking to increase their own benefit by getting more spectrum or some specific channels. This type of SSDF strategy may be collaborative or non-collaborative and magnitude of attack should be at least enough to nullify the robustness of global fusion rules.

2. Selfish SSDF nodes: Usually employed by dishonest nodes who are seeking to maximize their own benefit or looking to gain access to some specific candidate channels. Hence such dishonest nodes only lie on *channels of interest*. In such case, the attack is usually *collaborative*. A classic example of a collaborative selfish SSDF attack is a group of nodes belonging to a certain network provider which wants access to some selected channel(s); hence all nodes belonging to that provider send false information for those channels while truthfully reporting on all other channels in order to remain undetected. Magnitude of attack in such cases is *usually lower* than malicious nodes.

2.5.4 Random on-off attack

A malicious node may try to defeat the trust or reputation management algorithm by behaving cooperatively initially for a certain length of time and building its reputation. A period

of no attack is known as OFF period. Then the node would attack for a certain period of time known as ON period where magnitude of attack on each time slot is random. Then it would again behave cooperatively to regain any possible loss trustworthiness in the ON period. It is usually difficult to detect malicious nodes with such alternating behavior unless the trust management algorithms employs special measures.

CHAPTER 3: BACKGROUND AND RELATED WORK

In this chapter we discuss the state of the art research already done in the relevant field. First, we discuss salient concepts related to trust and reputation definitions in the literature. Then provide a comprehensive list of related research solutions regarding defense against spectrum sensing data falsification also known as byzantine attacks.

3.1 Background on Trust Reputation and Recommendation

The original concept of trustworthiness has its origins in human society known as social trust– a qualitative way of saying whether a person is reliable based on previous or current observations. This idea has been extended to the realm of distributed and networked systems where entities are analogous to humans and interactions exist between them. This idea has led to the concept of computational trust a quantitative measure of trustworthiness for interactions between entities in a network. The notion of trust is *contextual* and its meaning varies according to the applications and services they are associated with and hence quite challenging to define as it manifests itself in many different forms. In the following subsections, we present the commonly used definitions of trust and reputation.

3.1.1 Trust

Trust is can be broadly defined in a number of ways:

Reliability trust: *Trust is the subjective probability by which an individual entity A (trustor), expects that another entity B (trustee) performs a given action on which its welfare depends.* (*Gambetta [22]*). However, having high reliability trust on an entity B is not a sufficient condition to enter into a position of dependence on that entity (Falcone et.al. 2001). For example, it is possible that the value of damage in case of failure is too high to choose a given decision branch, and this independently either from the probability of failure (even if it is low) or from the possible payoff (even if its high). In other words, given the same trust value T, two different systems S and S' may view it differently as the associated risk may be too high for one particular system S enough to act as deterrent to enter into a state of dependence. On the other hand, another system S' may have less associated risk with it, in case of a failure and hence may agree to enter into a state of dependence. The amount of risk that a system is ready to take is often termed as *risk attitude*, and plays a key role in how trust metrics are interpreted for decision making. For this we need a broader concept of trust provided by decision trust.

Decision trust: *Decision Trust is defined as the extent to which an individual entity A is willing to depend on the information provided by another entity B, with a relative feeling of security even though negative consequences are possible.* In particular, it can be represented as a *particular expectation* with regard to the likely behaviour of other entities [22].

This definition is more generic than the previous definition and accommodates the relative contextual vagueness associated with interpretation of trust for different kinds of systems. In general, decision trust is often associated with a threshold, below or above which a decision on the fitness of an agent's input is decided.

3.1.2 Categories of formal trust metrics

Representation: This type of trust metrics assigns a trust value to each entity from a generic range of values on the real number axis bounded by the upper and lower limits. The lower the value the less trustworthy any entity is. Higher values indicate more trust. These values may be bounded between $+100$ and -100 [23], 0 and 1 [24], or -1 and $+1$ [25]. The values can be discrete or continuous.

Subjective probability: Trust is expressed as a subjective probability between 0 and 1 and the assigned values are continuous. This implicitly means that the probability that an trustee will perform an action that is beneficial or at least not detrimental to trustor is high enough for us to consider engaging in some form of cooperation. Such a probability is usually conditioned over the data or evidence available, trustor's assessment of the situation, and distinct to the trustor in question; hence it is subjective. This probability is inspired from Bayesian statistics where a probability at a particular time is treated as prior for future update of the current posterior. Another property that accompanies subjective probability representation of trust for systems where trust evolves over time is the *Cromwell's rule*.

Cromwell's rule suggests the prohibition of assigning of an absolute 0 and 1 as trust because if the prior probability is 0 or 1, then according to Bayes' theorem, the posterior probability is forced to be 0 or 1 as well; no evidence, however strong, could have any influence on the posterior. This is improper for a system where Bayesian inference is used based on incremental evidence over time.

Logic of uncertain probabilities (Subjective Logic): This representation is the most popular for systems where some kind of ignorance or uncertainty exists. This model of trust representation is widely known as *Josang's belief model*. Trust is particularly relevant in conditions of ignorance or uncertainty with respect to unknown or unknowable actions of others. The condition of ignorance or uncertainty about other people's behavior is central to the notion of trust [22]. The logic of uncertain probabilities is an amalgamation of probability distributions and degree of uncertainty associated in monitoring the interactions in the system and the advantage is that this allows to incorporate ignorance or uncertainty in the analysis. Josang's belief model uses a quadruplet termed as *opinion* to express all components about the interaction with an agent namely trust, distrust, uncertainty and relative atomicity. Then an expected *opinion* is used as the expected trust value or belief given the uncertainty in evidence. An opinion includes the concepts of disbelief and ignorance in addition to belief itself [26]. Another advantage of this model compared to earlier representations is that it allows the trust computation module to control the expected opinion value (expected trust value or expected belief) depending on how a specific system views ignorance and how it decides uncertainty would contribute towards belief or disbelief. Mathemati-

cally opinion is represented as $\omega = (b, d, u, a)$ where the components represent degrees of belief, disbelief, uncertainty, and relative atomicity respectively; where $\{b, d, u\} \in [0, 1]$ such that $b + d + u = 1$, and the expected opinion $E(\omega)$ known as degree of trust incorporating contextual uncertainty is represented as

$$E(\omega) = b + au, \tag{3.1}$$

The relative atomicity $a \in [0, 1]$ decides the extent to which the degree of uncertainty contributes to $E(\omega)$. In cases of absence of any prior statistical information, a is the reciprocal of the total number of possible situations that can be modeled, also known as *frames of discernment*. For e.g., if we are trying to find whether a node in a network is malicious or not, there are two frames of discernment, hence $a = 0.5$ assuming no other information is available on that node. Similarly, if we were to deal with expectation of the outcome of rolling a dice, $a = \frac{1}{6}$, due to six possible frames of discernment in a die, given additional information is not available about the dynamics of this particular die.

3.1.3 Reputation

Unlike trust which typically models reliability between two parties, notion of reputation does not necessarily involve binary parties and can represent a multinomial nature of interactions with multiple parties. For. e.g., if we want to know the reputation of an entity the result may be good, bad, average, excellent, neutral– a notion typically different from the binary

concepts. The term reputation can be loosely defined as, "*what is generally said or believed about a persons or entity's character or standing.* It is a *combined measure of trustworthiness as seen by all other entities interacting with a particular trustee.* The concept of reputation is closely linked to that of trustworthiness, but it is evident that there is a clear and important difference [27]. The differences lie in their multinomial nature and the essence of a *long term collective measure* of trust. Notion of reputation is usually associated when nature of interactions, referrals or ratings about an entity are shared over a long period of time or multi hop information for the network is available. In a broadcast media where same information is injected to everyone, the reputation of an entity B as seen by all entities in its neighborhood should converge to the trust that exists between entity B and a particular entity A within its neighborhood. For the purpose of this work, we can summarize Reputation as the long term average trust value of an entity as experienced by the overall network.

3.1.4 Applying trust and reputation for cooperative decisions

The application of trust for a particular cooperative decision is a essentially a binary concept, in the sense that whether an entity is going to *accept or not accept*, the information provided by the other interacting entity. Trust is also used for classification problems that are binary in nature, viz. fit or unfit, malicious or not malicious, allowed or banned etc. Reputation on the other hand is not necessarily a binary concept and may be applied for a variety of uses viz. making policy changes, learning about strategies employed by different users, resource

recommendation and provisioning. Recommendation is usually given based on the trust and reputation value that indicates the fitness of an entity towards a decision or sorting entities in order of their quality of contribution to the system.

3.2 Literature of SSDF Attack Remedies

Let us discuss the various techniques proposed so far to deal with SSDF attacks.

3.2.1 CatchIt: Heuristic onion peeling approach

'CatchIt' is a technique that helps preserve the correctness of spectrum decision in collaborative spectrum sensing even in the presence of multiple malicious nodes [28]. This heuristic can be described as an "onion peeling approach", where the possibility of a node being malicious is calculated in a "batch-by-batch basis", i.e., suspicious levels of all nodes involved are calculated at every time slot, and if at some point the suspicious level is greater than a certain threshold then that node is deemed to be malicious. The centralized decision center excludes the information from that particular node. The process is repeated until there are no more malicious nodes. Another paper [29] almost similar to [28], discusses the suspicion levels of different nodes using a Bayesian detection approach that progressively eliminates users based on past reports. It also proposes a consistency value that provides stability to variations in the suspicion levels assuming the presence of only one malicious node.

3.2.2 Weighted sequential probability ratio test

Robust distributed spectrum sensing is a method to ensure that the final spectrum decision is not affected by byzantine attacks when multiple nodes participate in collaborative spectrum sensing in the presence of a centralized decision maker [8]. There are two issues that are considered for robust fusion. (i) Ensure bounds on both false alarm and missed detection probabilities and (ii) consider the previous history of behavior of individual sensing terminals. The first issue is taken care by a weighted decision variable derived from the WSPRT [30, 31], (originally known as Abraham Wald's SPRT) where the weight of the decision variable is a function of the reputation. The second aspect is taken care by reputation maintenance where the previous behavior of a terminal is incremented or decremented based on the decision variable. Weighted SPRT uses weights over decision variables to account for reputation based on observed behavior. The final decision depends on whether the weighted decision variable is within the tolerable limits of false alarm and missed detection probabilities. For each secondary node i, its reputation r_i is set to zero initially. The most recent sensing report sent to fusion center is u_i and the final decision at the fusion center is u. At any stage, the reputation of node i is updated as

$$r_i \leftarrow r_i + (-1)^{u_i+u} \tag{3.2}$$

The idea of the hypothesis step is derived from sequential probability ratio test where the following likelihood ratio is modeled as a classical sequential ratio decision variable as:

$$S_n = \prod_{i=0}^{n} \frac{P[u_i|H_1]}{P[u_i|H_0]} \tag{3.3}$$

where H_1 is the hypothesis that spectrum is occupied, H_0 is the hypothesis that spectrum is unoccupied and n is the number of sample observations. The traditional decision variable is then modified to include the previous reputation of nodes and the weighted decision variable W_n is calculated as

$$W_n = \prod_{i=0}^{n} (\frac{P[u_i|H_1]}{P[u_i|H_0]})^{w_i} \tag{3.4}$$

where weight w_i is a function of the reputation of node i; $w_i = f(r_i)$. The decision variable is then compared with two predefined thresholds to get a final decision. The thresholds denoted as η_1 and η_0 are functions of the tolerated false alarm probability and the tolerated missed detection probability. Comparison of W_n with these two thresholds determines the final robust decision u. If W_n is greater than η_1, the final decision is $u = 1$. If W_n is less than η_0, then $u = 0$. For all other cases, it is necessary to take another observation. Given that the probability of final decision being true is greater than 0.5, a sensing radio with more accurate local sensing report has a higher expected reputation value. Hence attackers with inaccurate spectrum results will have a lower reputation value. We observe that the Weighted SPRT in [8] and Wald's SPRT discussed in [30, 32], are based on the same concept,

but in this paper the concept is modified by putting a weight on the sequential probability ratio which captures the previous activities/reputation of a radio.

3.2.3 Abnormality detection using double sided neighbor distance algorithm

Catching attackers with the help of a technique popularly used in data-mining called the k-proximity algorithm has been proposed in [33]. This considers a single channel system with secondary nodes in presence of a central data fusion center and non-collaborative malicious nodes. The proposed algorithm finds outliers that lie far apart from most secondary users in the history space. If the history of behavior if too close or too far to other histories, then an aberrant behavior is inferred.

3.2.4 Two-tier optimal cooperation based secure spectrum sensing

A distributed spectrum sensing algorithm is presented in [34] that aims to mitigate each of the two types of attacks namely PUEA and SSDF attacks. For PUEA, a user verification scheme on localization based defense is proposed. For SSDF a non-linear cooperation scheme which considers $M-$ary hypothesis, where M is the number of primary transmitters, is proposed. As opposed to the works in [8, 33], this paper introduces the concept of a 2-tier hierarchal centralized cognitive radio network, in order to optimize the energy and bandwidth consumed as well as decrease the computational complexity. Since reporting by a large number of

secondaries results in high computation, energy, and management costs, such optimizations are necessary. Thus special relay nodes which collect and compress local spectrum sensing help reduce costs.

3.2.5 KL divergence based defense

In [21], an analysis of collaborative and non-collaborative Byzantine attacks derived from [35] is presented. The paper aims to analyze the optimal attack strategies as well as issues of collaborative byzantine attacks with a dedicated fusion center, where Kullback-Leibler divergence (KL distance) is used as an objective function which malicious nodes seek to minimize. Given the probabilities of missed detection, false alarm, and the probabilities of true reporting for honest as well as malicious nodes, the paper provides the optimal fraction of malicious nodes required to make the fusion center incapable of making a correct decision. The aim of the malicious nodes is to introduce an error in the global decision on spectrum occupancy. The probability distribution function for the event that fusion center decides the result (j=0/1) on the hypothesis that PU is present (or absent) is calculated and denoted as X_j(or Y_j). Both of them are functions of the fraction (α) of malicious attackers in the system. The relative entropy or KL distance is a non-symmetric measure of the difference between the two distributions X and Y and is denoted by: $D(X||Y) = \sum_{j\in 0,1} X_j log\frac{X_j}{Y_j}$. The attackers attempt to reach a state where $D(X||Y)$ is zero, which is achieved for the optimal fraction of attackers. Subsequently, the paper discusses the best possible strategy for all the

entities namely the Byzantine radios, honest radios and the fusion center. The interaction between them is modeled as a minimax game between Byzantines and fusion center and the best strategy for both players is the saddle point. The interaction is analyzed in light of two different performance aspects namely, the KL distance and probability of error. The saddle points in the context of KL distance for both independent and collaborative Byzantine attacks are derived.

3.2.6 Majority voting based exclusion with long term reputation

A method proposed in [36] counters Byzantine attacks over a number of sensing periods, by accumulating the local decisions from each radio, and comparing it with the final decision at the fusion center in the same time window. The number of times the local decision from a radio is different from the final decision at fusion center is used as a reputation measure for a radio. If the reputation measure is lower than a certain threshold the radio is isolated from the fusion process. The methodology assumes the usage of 'l-out-of-K fusion rule' where final decision on a channel is decided what at least l out of K participating radios advertise. However, if the fraction of attackers is high, the fusion center cannot distinguish correctly.

3.2.7 Bio inspired consensus based cooperative sensing scheme

In [37] a scheme that is derived from bio-inspired consensus algorithms is utilized for a consensus based cooperative sensing scheme in an ad hoc cognitive radio network to counter SSDF attacks or Byzantine failures. The lack of a central authority makes ensuring security difficult as certain local information when spoofed impacts the radio behavior rather easily. In this method, RF statistics from immediate neighbors are used as state variables which are aggregated to deduce a consensus variable. The consensus variable is then used to make the decision over the detected energy and determine the presence or absence of the primary. The sensing scheme works in the following fashion. All secondary nodes sense the spectrum and report their locally estimated energy level to their neighbors. With the gathered information, a node uses a selection criterion to exclude reports from nodes that are likely to be attackers. At any time instant, the exclusion/selection process uses the mean value of energy at the previous instant and compares the mean value with individual values from the neighbors. For a particular node, the set of neighbors whose reports suffer maximum deviation from the mean are excluded and the remaining nodes' reports are taken into consideration. This process of sharing, receiving, selecting, and updating continues until all states converge to a common value which is then compared with a certain threshold. If the common value is greater than the threshold, the spectrum is occupied else it is not occupied.

3.3 Motivation for this Work

Even though some advances have been made to deal with vulnerabilities in cooperative spectrum sensing there are certain deficiencies in the current body of research which motivate the ideas behind this work. Moreover, there is very little body of work for distributed networks where binary opinions (hard-decision) on channel occupancy are shared during cooperative sensing. Outlier detection or voting based techniques might not work well compared to blind non-filtered fusion if a node fusing reports has more malicious nodes in its vicinity. (i.e., a higher local density of malicious nodes). In such cases, the fusion center does not have the accurate knowledge of ground truth as shown [21, 36]. Such techniques are also less robust to collaborative SSDF attacks where malicious agree upon channels they falsify. Another disadvantage of current majority voting based exclusion model is that they might infer a legitimate node as malicious simply because it is outside the reception radius of the primary transmitter.

Other important issues related to location privacy requirements and node identities have not been adequately addressed while defending against SSDF. Also realistic physical environments and their effects over the monitoring and trust models have not been discussed. Most designs consider single channel systems, although multi- channel local sensing results are shared usually shared among nodes. Almost all the existing work considers sharing of raw energy values during cooperative sensing although in recent times, sharing of binary opinions for local channel occupancy has been emphasized due to increased control channel bandwidth

costs. Furthermore, past works do not take into account the temporal aspect of SSDF attacks or malicious nodes having different levels of aggression on a multi-channel system. Hence there is a need to address the problem of establishing trustworthiness, malicious node identification and robust information fusion under SSDF attacks that (i) work in a distributed CR network, (ii) share multi-channel binary occupancy vectors, (iii) preserve location privacy requirement, (iv) consider local density variations of malicious nodes to be high, (v) works for varying levels of aggressiveness of dishonest nodes, (vi) remain valid under varying pathloss environments, and (vii) address situations where all neighbors do not legitimately view the primary transmission equally.

CHAPTER 4: ANOMALY MONITORING TECHNIQUE FOR SSDF ATTACKS

This chapter proposes a node centric anomaly monitoring technique based on which trustworthiness of a particular node is computed. The proposed method is then extended to consider noisy control channels which potentially affects the local sensing reports for all nodes. Finally, we discuss an extension which is used for utilizing misleading information and learn about the strategies of malicious nodes.

4.1 System Model and Assumptions

In this section, we describe the system model and assumptions. We present the threat model and SSDF attack strategies.

4.1.1 System model

We assume that secondary nodes undergo spectrum sensing and determine whether a channel is occupied by primaries or not. A secondary node i constructs its observed occupancy vector as: $B^i_{act} = [b_1, b_2, \cdots, b_n]$, where b_k is 1 or 0 depending on whether the channel k is decided

as occupied or unoccupied and N is the number of channels being monitored. A decision is made by comparing the energy sensed on channel k with a common normalization threshold γ_{th}. If sensed energy is more than γ_{th}, channel k is marked as 1 signifying 'occupied' and vice-versa.

Once this binary vector is created, a secondary node would broadcast this information to its neighboring nodes. Similarly, a secondary node would also hear broadcast messages (binary occupancy vectors) from its neighbors. The cooperation of binary vectors takes place over a common control channel which may be on licensed or unlicensed band [5, 16]. Based on the received vectors, a node will employ a fusion technique (e.g., majority voting) to obtain a better estimate about the spectrum usage that can significantly improve the performance of spectrum sensing [6, 8]. Such cooperative sensing has other benefits such as mitigating the shadowing and multi-path effects. We assume all nodes transmit through a common control channel while advertising its binary vectors during cooperative sensing and hence have equal transmission power agreed upon for signals sharing binary vectors. We make the following assumptions:

- We consider an ad-hoc secondary CR network with N nodes with γ_{mal} fraction of nodes being malicious; H denotes the set of honest nodes and M the denotes the set of malicious/dishonest nodes. The secondary network has no dedicated central fusion center, and each individual node fuses the spectral sensing data it receives from other nodes from which it can hear from and forms its opinion on the availability

of spectrum. The nodes then use any collision aware channel access framework for accessing spectrum from the pool of available channels.

- Nodes need not be aware of the geographical coordinates of other nodes which addresses location privacy demands and eliminates possibility of location falsification. Moreover, in a heterogeneous network with different operators, maintaining location information is an added overhead.

- Transmit power for advertising local sensing reports is equal. This assumption is valid since sensing reports are sent over a common control channel with each node having a fixed *sharing radius*. Knowledge of the transmitter output power, channel losses, and antenna gains with the appropriate path loss model allows us to find distance between the two nodes using Received Signal Strength (R.S.S) through R.S.S lateration [38, 39]. We only harness the distance information for our work and not location information.

- Each primary transmitter *whether it chooses to transmit or not*, transmits only on one channel; so the channel associated with a primary transmitter is known. The primary transmitter is assumed to be a transmitter like TV tower or a stationary cellular base station. For cellular primary network, this approach is applicable for downlink primary channels. This is because in a cellular network, the source for the uplink channels may be mobile. In such cases, there has to be location information available for such mobile primary receivers. However, most of our work is associated with TV towers or downlink channels of wireless base stations.

- Primary transmitter that transmits on channel k, is referred as T_k, and since it is fixed, its coordinates of a primary (x_{T_k}, y_{T_k}) are known to the nodes.

- Outcome of sensing at a node are raw energy values which are converted into a binary vector of 0's and 1's, where 0 represents absence of primary and 1 represent primary's presence. This is known as hard-decision sharing of cooperative spectrum data.

- There is negligible noise between two secondary neighbor nodes while transmission of binary occupancy vectors over the control channel.

- The nodes use a majority voting fusion to fuse all binary vectors from its neighbors within a sharing radius. Unlike in [33], which discusses a very restrictive fusion model (AND fusion rule), we use majority voting fusion rule, which gives better flexibility towards errors committed by nodes and/or malfunctioning nodes. This approach has been proved to effective in many voting based defense research.

- We do not assume that all neighbors of a node are necessarily within the coverage area of primary transmitters.

- The probability of false alarm is the probability that a channel which is actually empty (H_0) is erroneously detected by a node to be occupied, and is denoted by P_f or $P(H_1|H_0)$. Similarly, the probability of missed detection is the probability that a channel which is occupied (H_1) is not detected by a node and is denoted by P_m or $P(H_0|H_1)$. These are traditionally for the channel between the primary and the CR node. There is body work that deals with calculation of such probabilities [40].

4.1.2 Threat model

Malicious nodes may have *different levels of aggression*. A very aggressive attacker risks itself to easier detection, while a less aggressive attacker hardly effects the network in a significant manner. Thus the level of aggression is often a choice between how much time a malicious node wants to remain undetected, while inflicting maximum damage to the other nodes. Intuitively, a smart attacker may not attack on the same number of channels or the same channel sets on all time slots. The level of aggression of malicious nodes which we call *magnitude of attack* is reflected by the value of the proposed attack measures discussed next.

4.1.3 Attack measures

- Probability of Attack (P_{attack}): A malicious node falsifies report for a random number of channels every time slot, and channels falsified upon are also random. However, the nodes follow a long term mean number of channels that are attacked. The value of this mean is denoted as P_{attack}, which depends on how aggressive the malicious node is. Hence for a system of N channels, $P_{attack} = 0.70$ means that the probability of any channel occupancy status being altered is 0.70.

- Intensity of Attack (I_{attack}): A malicious node falsifies the report for a *fixed number* of channels every time slot. However, channel IDs falsified on are *randomized every*

time slot. The fraction of channels falsified on every time slot is defined as *intensity of attack* and denoted by I_{attack}.

If the nodes want to play a perfect mixed strategy in a anti-coordination game, the optimal attack strategy is when I_{attack} or P_{attack} is 0.5. More aggressive attackers have attack magnitude greater than 0.5 and more conservative attackers have attack magnitude less than 0.5. It may be noted that very low magnitude do not affect the network significantly while extremely high magnitude facilitates easy detection; hence for practical purposes we ignore the two extremes. We also subject our system to both types of attack and observe their effects to report which is a better attack strategy from the attackers perspective. A smart malicious attacker will randomize attacks on different channels and different number of channels.

4.1.4 Attack strategy: Collaborative and non-collaborative SSDF

- Malicious nodes are said to launch a collaborative SSDF if such nodes also agree upon the channel IDs they decide to falsify. The advantage of such a strategy is that the probability of compromising the voting based fusion rule increases. For example, if a node has 5 neighbors out of which 3 are malicious falsifying on the same channel, a majority voting based fusion rule is bound to fail. For collaborative SSDF, the member channels of the channel set chosen for attack on each time slot is the same, although the set may itself change over time. The channel set is a subset of the total number of channels in the operating spectrum. The disadvantages of a collaborative SSDF is that

it may be costly and time consuming and may not return good results if two nodes collaboration do not sense the same channel status on a channel it intends to falsify. If the channel set remains same over time, it is called static collaborative attack, else it is called dynamic collaborative attack.

- In independent SSDF attacks malicious nodes do not collaborate on the channel sets they falsify.

Our node centric anomaly monitoring mechanism is not dependent on the type of strategy employed. The proposed method works well regardless of the SSDF strategy employed unlike other works discussed in the related work.

4.2 Anomaly Monitoring Mechanism

In traditional wireless ad-hoc networks, examining packets forwarding is a common method for monitoring aberrant behavior and track malicious or selfish nodes for trustworthy and secure routing [41]. The successes and failures of such interactions are treated as evidence and is subsequently used for trust modeling or identifying possible malicious intent. In the context of cooperative spectrum sensing in CR networks, we need to monitor anomalies in spectrum sensing data reported among the nodes participating in cooperation. Such anomalies can reveal incidence of spectrum data falsification attacks. The presence of such anomalies forms the basis of identifying the possible dishonest/malicious nodes, building trustworthiness, and then finally making the sensing mechanism more secure and robust.

We pursue this by *predicting the bounds on received power levels* over a channel for a particular neighbor node and then applying a *normalization criterion* over predicted bounds to obtain a predicted vector for occupancies. Each node calculates a predicted occupancy vector for its neighbors. Then we *compare predicted occupancy with the vector actually advertised by a neighbor*. Any mismatches between the predicted bounds on occupancy and the actual advertised vector is recorded as an event of an anomalous or non-cooperative behavior. The event where bounds on prediction matches the advertised vector is a recorded as occurrence of a match. The relative frequency of such matches (or relative absence of anomalies) given the total number of interactions on multiple channels over several time slots is a measure of how much trustworthy a node is. It is obvious that if the relative frequency of anomalies are higher subject to appropriate conditions, the trustworthiness of that node will be less and vice-versa. The formulation of gathering the evidence or observation data for trust computation is divided into the following parts: predicting bounds on received power, normalization criterion to map bounds into predicted occupancy vector, and comparison of predicted occupancy vector with advertised vectors. The comparison yields a data set which we call as *trust evidence* or *observation counts*. These terms are used interchangeably. A comprehensive list of notations used in this chapter is listed in Table 4.1.

Table 4.1: Notations

Symbol	Meaning
N^i	Neighbor set of node i
H	Set of honest nodes
M	Set of malicious nodes
γ_{th}	Common threshold used to normalize power vectors
$s_{T_{ik}}$	Distance between node i and primary tower T_k for channel k
b_k	Binary Decision on a channel k, $b_k \in 0, 1$
j	Set of all neighbors of i, $j \in N_i$
P^i	Measured power vector on n channels at node i
B^i_{act}	Actual binary occupancy vector formed at i
B^i_{adv}	Advertised binary occupancy vector by node i
$P^{ij}_{predict}$	Vector of power ranges for neighbor j predicted by i
B^j_i	Binary occupancy of node j, predicted by i
$b^j_k\|_{infer}$	Predicted decision on any channel k, for B^j_p
$(\alpha, \beta, \mu)^j$	Three tuple trust evidence
$E_{j,i}$	Trust of neighbor j calculated by node i
TBF^i	Fusion result Based on selective inclusion of j based on trust

4.2.1 Predicting bounds on received power

Suppose node i measures the power vector

$$P^i = \{\gamma^i_1, \gamma^i_2, \cdots, \gamma^i_n\},$$

where γ^i_k is the power received (sensed) on channel k and n is the total number of channels. Each node i forms its binary occupancy vector $B^i_{act} = \left[\ b^i_1, b^i_2, ...b^i_n\ \right]$ from it's power vector P^i by comparing γ^i_k with threshold γ_{th}, where

$$b^i_k \begin{cases} = 1 & \text{when } \gamma^i_k \geq \gamma_{th} \\ = 0 & \text{when } \gamma^i_k < \gamma_{th} \end{cases} \tag{4.1}$$

B_{act} is the actual binary report constructed by an individual CR node using a common network threshold γ_{th}. Each node i, advertises a public binary vector B^i_{adv} such that,

$$B^i_{adv} \begin{cases} = B^i_{act} & \text{if node } i \in H \\ \neq B^i_{act} & \text{if node } i \in M \end{cases} \tag{4.2}$$

where H and M denote the sets of honest and malicious nodes respectively. Just like node i advertises its binary vectors to others, it also hears similar advertisements of binary occupancy vector from its neighbors. For a neighboring node $j \in N^i$, node i estimates the bounds on possible received power on all channels using their mutual distances, its received signal strengths and known location of fixed primary towers. The mutual distance between the node i and its neighbor node j, can be calculated using received signal strength (RSS) localization or lateration [38, 39]. Assuming a generic propagation model for path loss, it is trivial to find the distance of a transmitter node given knowledge of transmit output power, antenna gains, transmit-receive side losses when the transmit power levels are same. The assumption on having equal transmit power while sharing binary vectors is reasonable because a broadcast message on spectrum data is shared within a common sharing radius through a control channel. Hence the transmit power for sharing vectors is same for all nodes. We use the following generic model for received signal in [39]:

$$RX_{pwr} = TX_{power} - PL_{Tx-Rx} + \Omega \tag{4.3}$$

where, RX_{pwr} is the received signal strength when a potential neighbor transmits with known power TX_{power}. Given the knowledge of transmit power, losses, and gains represented by Ω, we calculate the pathloss PL_{Tx-Rx}. We plug-in this value in Eqn. 4.4, and solve for d to find the mutual distance.

$$PL_{Tx-Rx} = PL_{1meter} + 10log(d^{\omega}) + f \tag{4.4}$$

where d is the distance between transmitter (neighbor j) and receiver (node i) which needs to be found; f is a constant for shadow fading; PL_{1meter} is the near field reference power and ω is the pathloss factor. The calculated PL_{Tx-Rx} from Eqn. 4.3 can be used in Eqn. 4.4 to get the desired estimated distance between the receiver and its transmitter. This distance is the distance between node i and its neighbor j which transmits binary occupancy vector, and is denoted by s_{ij} such that $d = s_{ij}$.

The distance s_{ij} allows us to plot a circular area around the location of the monitoring receiver i, with radius of s_{ij}. This circle is the locus of node j, and the neighboring node's location can be anywhere on this circle. We draw a straight line from the center of the circle to the primary transmitter T_k located at (x_{T_k}, y_{T_k}) as shown in Fig. 4.1. Using geometry it is easy see that point H is the closest distance and point L is the largest farthest from (x_{T_k}, y_{T_k}) to the circular locus of j. Under such conditions, the RSS due to T_k will be maximum on the point of the circle that is closest to T_k, i.e., on point H and minimum at a point L that is farthest from T_k. We denote the power levels at these two locations as $[\gamma_k^j]_{high}$ and $[\gamma_k^j]_{low}$ at distances $s_{j_{min},k}$ and $s_{j_{max},k}$, respectively. For all locations on the circle whose distances

are between $s_{j_{min},k}$ and $s_{j_{max},k}$, the RSS on channel k is bounded between $[\gamma_k^j]_{high}$ and $[\gamma_k^j]_{low}$. This forms our bounds on received power on a channel k for a particular neighbor j.

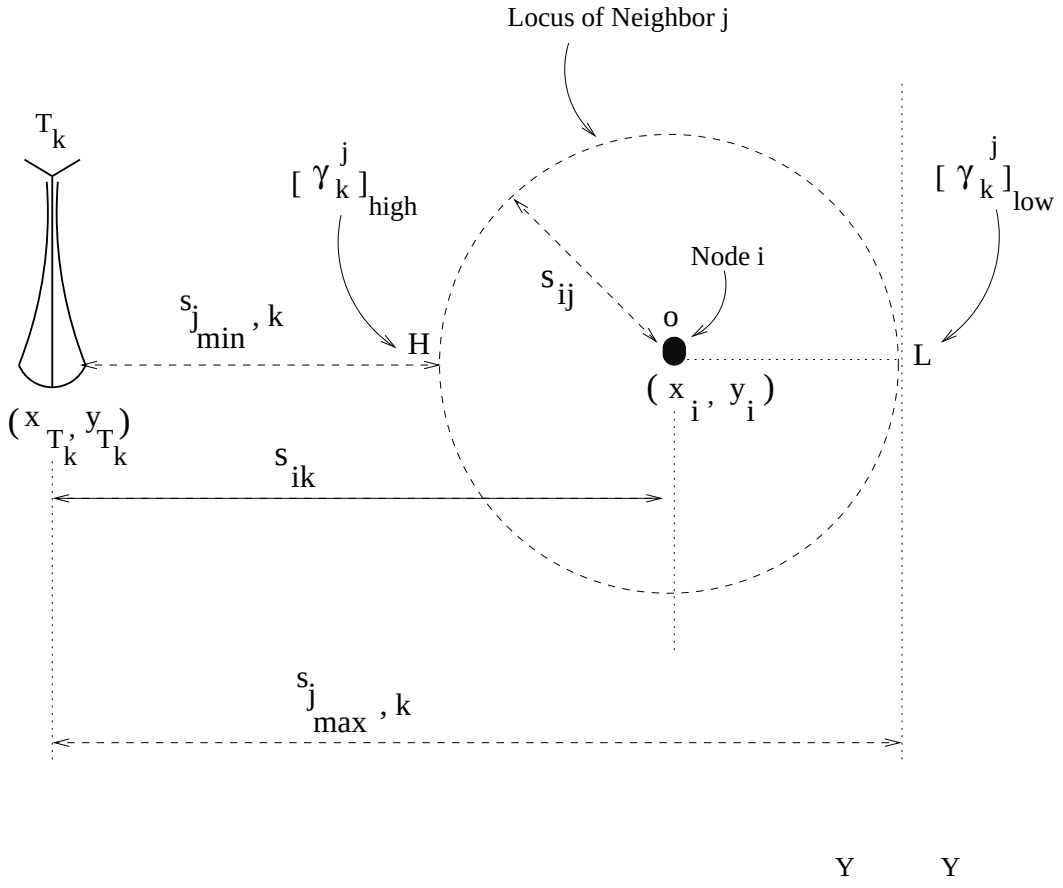


Figure 4.1: Maximum and minimum RSS on channel k for neighbor node j

The calculation of $[\gamma_k^j]_{high}$ and $[\gamma_k^j]_{low}$ is done as follows: Using commonly used models for modeling the primary signal propagation [42, 43, 44], we have

$$\gamma_k^i = P_k \times \frac{A^2}{s_{ik}^{\omega}}; \tag{4.5}$$

where, γ_k^i is the sensed energy detected on channel k at node i for a primary transmitter T_k, A = frequency constant, ω is path loss factor, s_{ik} is the distance between primary tower T_k and node i, and P_k is the transmit power of T_k. Usually, $A = \frac{\lambda}{4\pi}$ where λ is wavelength of light. From Eqn. 4.5, we get P_k which is used to get the bounds on possible received power due to the primary's transmission (see Fig. 4.1) as:

$$[\gamma_k^j]_{high} = P_k \times \frac{A^2}{s^{\omega}_{j_{min},k}}; \tag{4.6}$$

$$[\gamma_k^j]_{low} = P_k \times \frac{A^2}{s^{\omega}_{j_{max},k}}; \tag{4.7}$$

Now we divide the Eqn. 4.5 by Eqn. 4.6 and Eqn. 4.7 to find $[\gamma_k^j]_{high}$ and $[\gamma_k^j]_{low}$, respectively, since s_{ik}, $s_{j_{min},k}, s_{j_{max},k}$ and γ_k^i are known to node i. Node j may be anywhere on the circular locus. Thus the predicted power vector of node j is a 2-tuple vector

$$P^{ij}_{predict} = \left[([\gamma_1^j]_{low}, [\gamma_1^j]_{high}), ([\gamma_2^j]_{low}, [\gamma_2^j]_{high}), ..., ([\gamma_n^j]_{low}, [\gamma_n^j]_{high}) \right]$$

For most cases, the distance between primary and a secondary node is larger than the distance between two secondary nodes which are neighbors. However, in the unlikely event that a primary transmitter is in close proximity to a secondary CR and its neighbor, the calculation of $s_{j_{min},k}$ and $s_{j_{max},k}$ is slightly different. Such a situation is depicted in Fig. 4.2. In such a case, the line joining the node i and the location of primary yields distance s_{ik}. This line when further extended in the same direction meets the locus of neighbor j at point

H which is the nearest location on the locus from (x_{T_k}, y_{T_k}). This yields $s_{j_{min},k}$ and the corresponding lower bound of received power $[\gamma_k^j]_{high}$ is calculated using Eqn. 4.6. The same line when reflected 180 degrees will meet the locus of neighbor j at point L which forms the corresponding furthest distance yielding $s_{j_{max},k}$. This distance is used to calculate $[\gamma_k^j]_{low}$ using Eqn. 4.7.

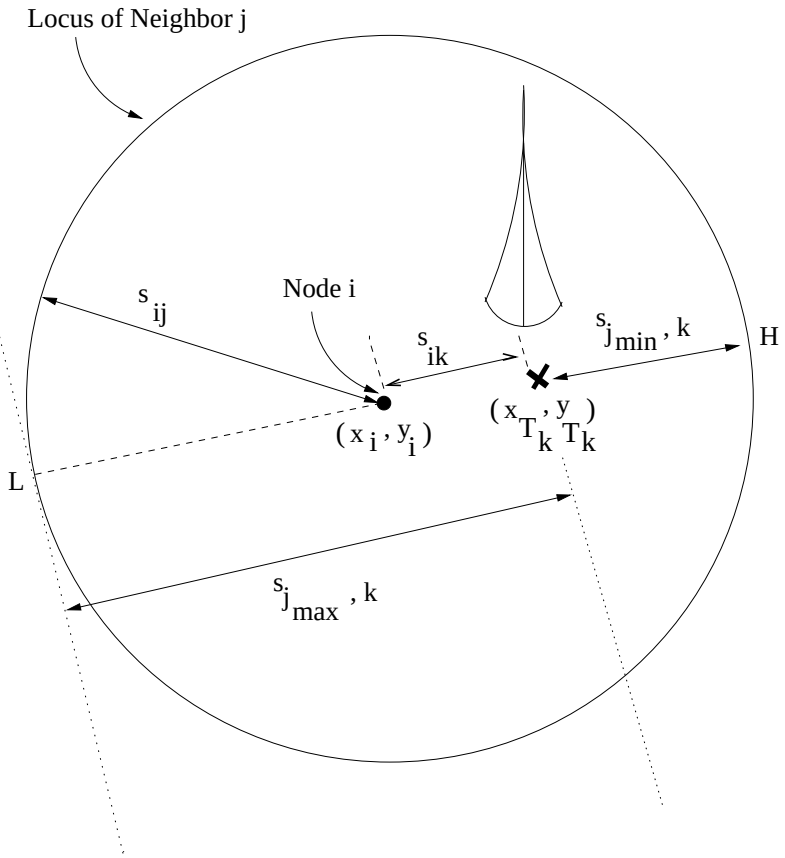


Figure 4.2: Calculation of maximum and minimum RSS on channel k for neighbor node j when T_k is between located between i and j

4.2.2 Normalization criterion for predicted occupancy

With the estimated power vector known, the inference drawn by i for neighbor j on channel k is,

$$b_k^j|_{infer} = \begin{cases} 0 & \text{if both } [\gamma_k^j]_{low} \text{ and } [\gamma_k^j]_{high} \leq \gamma_{th}; \\ 1 & \text{if both } [\gamma_k^j]_{low} \text{ and } [\gamma_k^j]_{high} \geq \gamma_{th}; \\ X & \text{otherwise} \end{cases} \tag{4.8}$$

where X denotes that no inference could be drawn. Eqn. 4.8 gives the normalization criterion. When both the lower and higher predicted power levels on a channel are less than normalizing threshold γ_{th}, it implies that channel k is not being used by any primary transmitter, i.e., channel is unoccupied. So in this case $b_k^j|_{infer}$ is inferred as 0. When both the lower and the higher predicted power levels on a channel are more than normalizing threshold γ_{th}, it implies that channel k is being used by a primary transmitter, i.e., channel is occupied. In such a scenario, $b_k^j|_{infer}$ is inferred as 1. In the event, where one power level is higher and the other is lower than γ_{th}, no inference can be drawn on that channel at hence it is marked as X. Usually we have to wait for future observations for a concrete inference on that channel.

Though our discussion with respect to single node i, the analysis is applicable to all other nodes as well. Hence we drop the suffix i from our notations.

4.2.3 Formation of trust evidence

The predicted occupancy vector, given the mutual distance between nodes i and j, is given as

$$B^j_{pre} = [b^j_1|_{infer}, ..., b^j_n|_{infer}]; \quad b^j_k|_{infer} \in 0, 1, X \tag{4.9}$$

Now node i compares $B^j_{pre} = [b^j_1|_{infer}, ..., b^j_n|_{infer}]$ with received $B^j_{adv} = [b^j_1, ...b^j_n]$ on corresponding channels for matches and mismatches. The comparison of B^j_{pre} with received $B^j_{adv} = [b^j_1, .., b^j_k..b^j_n]$ for each channel is done and the results are recorded using the criterion in Eqn. 4.10. After this comparison, all matches are denoted as α, mismatches are denoted as β, and channels with value X in B^j_{pre} are recorded as μ denoting undecided. If Q^j_k is the result of the comparison, then

$$Q^j_k = \begin{cases} \alpha^j & \text{if } b^j_k|_{infer} = b^j_k; \\ \beta^j & \text{if } b^j_k|_{infer} \neq b^j_k; \\ \mu^j & \text{otherwise} \end{cases} \tag{4.10}$$

The total number of matches, mismatches and undecided for each node j is denoted as η_{α_j}, η_{β_j} and η_{μ_j} such that $\eta_{\alpha_j} + \eta_{\beta_j} + \eta_{\mu_j} = n$. This 3 tuple vector forms the trust evidence. α_j is a treated as a positive rating, β_j is a negative rating, and μ_j is a neutral or uncertain rating. More number of positive ratings relative to the overall number of ratings indicates more positive behavior and vice-versa. The number of neutral ratings increases or decreases confidence on estimates as we discuss in Chapter. 5.

4.3 Fairness under Noisy Control Channels: A Special Case

We have to distinguish mismatches caused due to malicious behavior and mismatches that may be caused due to noise. Hence we propose a fairness model to discount the effects of mismatches that may be caused due to channel errors.

To account for channel noise, shadowing and fading, we define the probability of false alarm as $P_f = P(H_1|H_0)$, probability of missed detection as $P_m = P(H_0|H_1)$, and channel error probability due to noise as P_e.

P_f and P_m are due to inherent channel induced sensing inaccuracies when nodes are not able to detect the presence or absence of primary transmission. Moreover, when local sensed reports are advertised to the neighbors, they may be altered due to noise between two CR nodes. Considering the channel error probability, we define modified false alarm probability for node j at node i as

$$P_{fe}^{'} = (1 - P_f) \cdot P_e + P_f \cdot (1 - P_e)$$

and modified missed detection probability, as

$$P_{me}^{'} = (1 - P_m) \cdot P_e + P_m \cdot (1 - P_e)$$

$P_{fe}^{'}$ is the probability that a 0 in node $j's$ advertised vector will reach as 1 at node i, irrespective of malicious behavior of node j. Similarly, $P_{me}^{'}$ is the probability that a 1 in node $j's$ advertised vector will reach as 0 at node i, in spite of any malicious behavior. We

need to discount these mismatches caused by P_m, P_f and P_e to achieve a fair trust coefficient. Let the actual number of $0's$ and $1's$ in ideal case for any received vector be $x_0^{ideal_j}$ and $x_1^{ideal_j}$ respectively. Let the number of $0's$ and $1's$ in received vector from j be $H(0)^{received}$ and $H(1)^{received}$, which are known. Therefore,

$$x_0^{ideal_j} - P_{fe}^{'} \times x_0^{ideal_j} = H(0)^{received} \tag{4.11}$$

$$x_1^{ideal_j} - P_{me}^{'} \times x_1^{ideal_j} = H(1)^{received} \tag{4.12}$$

From Eqn. 4.11 and Eqn. 4.12, we find $x_0^{ideal_j}$ and $x_1^{ideal_j}$, the other parameters being known. The total number of mismatches from ideal scenario caused due to channel uncertainties and noise is

$$P_{fe}^{'} \times x_0^{ideal_j} + P_{me}^{'} \times x_1^{ideal_j} = \alpha_{noise}^{j} \tag{4.13}$$

where α_{noise}^{j} accounts for the mismatches that occur due to unreliable channels conditions. This α_{noise}^{j} will additionally play a role alongwith $\eta_{\alpha_j}, \eta_{\beta_j}, \eta_{\mu_j}$. We argue that under noisy channels and when the channel statistics are known, a deviation of α_{noise}^{j} on average is expected from the ideal case.

4.4 Special Monitoring Extensions for Intelligent Decision Making

In this section we propose certain extensions to the already discussed monitoring mechanism that help us in making intelligent operational decisions. We propose to record the channels

on which mismatches occur for neighbor j as a vector known as *Observed Invert Sequence* and denote it by $IS^{j,i}$. This would throw light on important information about learning whether there is pattern in the attacks from two discrete nodes, or if there is preference on certain channels that are attacked more. Such information can aid in understanding the rationale of attackers. Moreover, we use this vector to propose a method where we can *utilize misleading and spurious spectrum data* to the advantage of the CR network. For this, the only modification we need to make is in the Eqn. 4.10, where the channels on which β is encountered is added to the set IS_i^j.

$$Q_k^j = \begin{cases} \alpha & \text{if } b_k^j|_{infer} = b_k^j; \\ \beta & \text{if } b_k^j|_{infer} \neq b_k^j \text{ and } IS_i^j \leftarrow k; \\ \mu & \text{otherwise} \end{cases} \tag{4.14}$$

The Eqn. 4.14, is an extension that would help in inclusive learning based approach towards robust and secure network operations in CR networks.

4.5 Summary

In this chapter, we proposed an anomaly monitoring technique for a distributed CR network with malicious byzantine nodes that launch SSDF attacks. The anomaly monitoring technique builds trust evidence or observation counts. This trust evidence has three types

of observation namely a match, mismatch and undecided. Our proposed model will be used in the following chapters for computing trust of nodes. We also distinguished between mismatches that may be caused due to channel noise as opposed to mismatches introduced due to malicious behavior. Finally, we proposed the concept of invert sequence vectors capturing channels on which mismatches are detected, for subsequent intelligent operational decisions.

CHAPTER 5: TRUST MODELS

In this chapter, we propose models for computing trust and reputation of nodes based on the observed evidence from the anomaly monitoring phase. We incrementally build our trust models from the simplest to the most advanced depending on the potential requirements and objectives of the network. The first model proposed is for a normal CR system that is risk neutral. This model known as *Optimistic Trust Model* (or Model I), is useful when the focus is not on identification or isolation of malicious nodes but only filtering out spurious sensing reports from featuring in spectrum data fusion. We also propose a modification of the optimistic trust model through evidence coarsening and show that it approximates the expectation of a beta distribution. The advantage of such an approximation is that while giving similar trust values as optimistic trust model, the other moments like variance, confidence intervals, skewness etc can be easily calculated. It also helps in confirming with some established trust metrics. Finally, we propose a *Dirichlet distribution framework* based trust model (Model 2) that is applicable for a conservative system which is risk averse. The difference between Model 1 and Model 2 is largely in the way uncertainty in evidence is interpreted.

5.1 Issues with Quantifying Trust and Reputation in CR systems

For malicious node identification, it is important that entities are long lived, such that sufficient incremental evidence is available. This is more important where the malicious entities have some form of randomness in their attack strategies. Malicious node identification or node isolation is not feasible in highly mobile distributed network as neighbors of a node keep changing or nodes enter and exit the network frequently. Node identification makes sense for a static network or a network with limited mobility. For other cases, robust fusion is more natural choice for defense where trust values are used to disregard information from possible outliers. Given the few number of observations from a particular entity, characterizing its fitness to the network by labeling it as honest or malicious is not feasible. Since cooperative sensing only concerns immediate neighbors, multi-hop trust propagation popular is not required. A trust enforcement infrastructure is usually necessary for node isolation decisions.

5.2 Optimistic Trust Heuristic

To trust or not to trust is a binary concept unlike reputation. Hence we need to map a ternary evidence into a binary notion. The 'event' of observing a 'match' is conceptualized as achieving trustworthiness and a mismatch is treated as a failure to achieve trustworthiness. To account for the number of channels as X where no inference could be drawn, we split

them in the ratio $\eta_\alpha : \eta_\beta$ and add to the number of matches. This assumption is valid because we assume that there is no preference on channels attacked and channels selected for attack are *uniformly random*. Thus the proportion of matches for node j is updated as $\eta_{\alpha^j} + \frac{\eta_{\mu^j}}{\eta_{\alpha^j}+\eta_{\beta^j}} \times \eta_{\alpha^j}$.

The proportion or relative frequency of matches to the total number of channels can be treated as the instantaneous trust value for node j as computed by node i and is given by

$$E^{j,i} = \frac{\eta_{\alpha^j} + \frac{\eta_{\mu^j}\eta_{\alpha^j}}{\eta_{\alpha^j}+\eta_{\beta^j}}}{\eta_{\alpha^j} + \eta_{\beta^j} + \eta_{\mu^j}} \tag{5.1}$$

where $0 \leq E^{j,i} \leq 1$.

The value $E^{j,i}$ is always a value between 0 and 1. Values closer to 1 indicate more trustworthiness and lower values indicate relatively less trust. In general, this process is repeated over time; hence $E_t^{j,i}$ is the trust calculated based on evidence collected on time slot t.

5.2.1 Computing bounds on trust values and certainty

We analyze the bounds on computed trust values to show how *confident* we are about the trust value given by Eqn. 5.1. The computed bounds on trust values provide a confidence

heuristic called *certainty*. This becomes particularly important when number of undecided is high.

The number of μ^j's can have any number of matches or mismatches which is unknown to the monitoring node. The trust attains a *maximum value if all* μ's *were matches* and a *minimum value when all* μ's *were mismatches*. Hence $E^{j,i}_{high} = \frac{\eta_{\alpha^j}+\eta_{\mu^j}}{N}$ and $E^{j,i}_{low} = \frac{\eta_{\alpha_j}}{N}$ respectively for the maximum and minimum case. It is not difficult to see that this interval $E^{j,i}_{low}, E^{j,i}_{high}$ depends on how large μ^j is. The larger this interval, the lower the probability of the true relative frequency to be closer to the expected (trust) value and hence lower the confidence. How large η_{μ^j} is, depends on a number of factors like pathloss and the relative spatial orientation of node pairs to be discussed in Chapter 6.

The chances of the extreme cases that all undecided were all mismatches or matches over long time is low as there are no preference on the channels attacked. Also, when there is a change in position of neighbor, uncertainty may be reduced. Since the channels to be attacked are completely random, the assumption that μ's are split in ratio of recorded η_{α^j} and η_{β^j} is justified. Nevertheless, the condition ($E^{j,i}_{low} \leq E^{j,i} \leq E^{j,i}_{high}$) always holds and the interval can be used to sort nodes according to trustworthiness.

5.2.2 Trust and certainty measure

Since the larger the interval $\delta = (E^{j,i}_{high} - E^{j,i}_{low})$, the lesser is the confidence, we use $1 - \delta$ as the certainty metric that defines how much confident we are about $E^{j,i}$. In cases where

number of undecided are less, there are more matches or mismatches which make the opinion about the trust more certain. Hence when we have to assign a trust of node j, where perfect information is not present, we use *certainty* to indicate how confident we are about the evaluation of $E^{j,i}$. We define certainty as $a^{j,i} = 1 - \delta$. The trust-certainty tuple for neighbor j is represented as $(E^{j,i}, a^{j,i})$.

Table 5.1: Trust-Certainty tuple; N=40

Scenario	α	β	μ	**Trust,Certainty**	**Beta Trust**
1	14	13	13	$0.51, 0.675$	0.50
2	19	18	3	$0.51, 0.925$	0.50
3	22	0	18	$1.00, 0.55$	0.97

5.2.3 An illustrative example

Let us consider the three scenarios as shown in Table. 5.1. If we compare the first two scenarios, we see that both of them have the same trust but Scenario 2 has more certainty or the confidence of the trustworthiness because true observations are known on 37 out of the 40 channels. Scenario 2 has $E^{j,i}_{low} = 19/40$ and $E^{j,i}_{high} = (19 + 3)/40$, hence bounds are $(0.475, 0.55)$ and $\delta = 0.075$ making the interval δ smaller than scenario 1. Hence $a^{j,i} = 1 - \delta = 0.925$ which is higher than scenario 1. Thus, scenario 2 will be more confident and hence have more priority than scenario 1 even if they have same trust value. In all other cases, the trust value has the priority in determining relative trustworthiness. Hence scenario 3 is most trustworthy followed by 2 and 1.

5.2.4 Trust evidence coarsening into a modified parameter Beta distribution

Ternary evidence can be alternatively modeled by coarsening it into a binary space [45] to make it mathematically tractable with known distributions like the Beta distribution [46]. Beta distribution is used widely for trust modeling where state of evidence is binary. Beta distribution suggests that if r is the number of positive and s is the number of negative outcomes of an experiment, then it can be modeled as a Beta distribution with parameters specified by $\varphi = r + 1$ and $\kappa = s + 1$ with the posterior probability density function given as

$$f(p|\varphi, \kappa) = \frac{\Gamma(\varphi + \kappa)}{\Gamma(\varphi)\Gamma(\kappa)} p^{\varphi - 1}(1 - p)^{\kappa - 1} \tag{5.2}$$

However, Eqn. (5.2) is of less physical significance for representing trustworthiness [46]. The mean of the pdf in Eqn. (5.2) can accurately model trust metrics given by

$$E = \frac{\varphi}{\varphi + \kappa} = \frac{r + 1}{r + s + 2} \tag{5.3}$$

Following this, we can treat the floor of the numerator of Eqn. (5.1) as the coarsened number of matches or positive outcomes denoted as α_j^c and the $N - \alpha_j^c = \beta_j^c$ as the coarsened number of mismatches or negative outcomes. Given this, the trust value can be modeled as the expectation of a beta distribution with parameters $(\alpha_j^c + 1, \beta_j^c + 1)$ given as

$$E_{beta}^{j,i} = \frac{\alpha_j^c + 1}{\alpha_j^c + \beta_j^c + 2} \tag{5.4}$$

where $\alpha_j^c = \lfloor \eta_{\alpha_j} + \frac{\eta_{\mu_j}\eta_{\alpha_j}}{\eta_{\alpha_j}+\eta_{\beta_j}} \rfloor$. We can observe from Table 5.1, that Eqn. (5.1) and Eqn. (5.4), almost give the same value. This observation holds true as we assume the channels chosen for attack are uniformly random. The advantage of using a Beta distribution inspired expectation is that we can easily calculate all the other moments and error margins and confidence intervals are standard and known. Also the Beta trust value adheres to the Cromwell's rule by preventing the trust value to be assigned as absolute values of 0 and 1. Infact for all exhaustive combination of values between 1 and N, it gives approximately equal results to optimistic trust heuristic except for boundary cases where there is total absence of a particular observation (say zero mismatches). The coarsening is usually not recommended if a system has high uncertainty and associated risk due to failure is high.

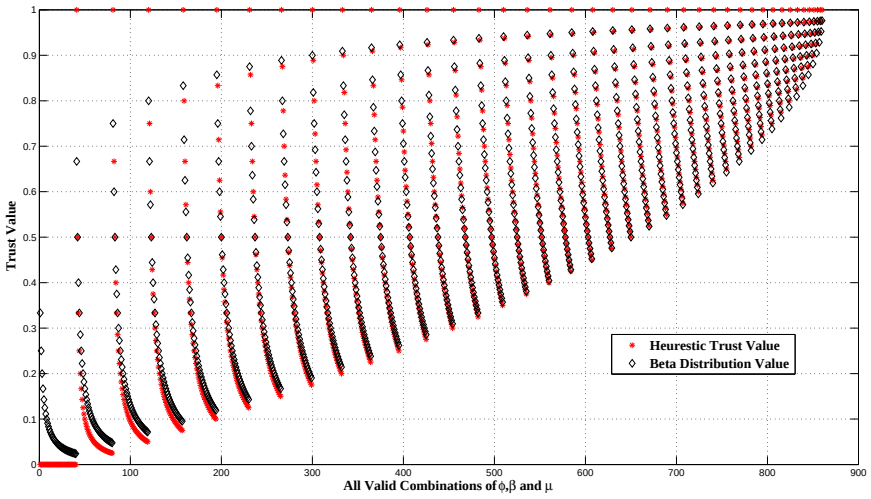


Figure 5.1: Coarsened Beta Trust Value vs Relative Frequency Trust

5.2.5 Motivation for a more pessimistic Dirichlet recputation based trust

In the design of a more defensive or pessimistic system with a higher risk attitude, splitting the undecided in the ratio of observed matches and mismatches does not have scientific basis for nodes with a high number of undecided ratings. For a conservative CR system the assumption of uniform preference on channels is improper due to the following reasons:

1. Consider a case when channels attacked with some preference on certain channels: some are attacked more, while others are attacked less. An intelligent adversary might employ a variety of statistical techniques for its attacks. Hence the channels attacked are not uniformly random.

2. Furthermore, even if statistical preference is not employed there is a non-zero probability that all channels which were attacked on were inferred as undecided. Such a possibility is high when there are a high number of undecided observations in the trust evidence and channels that were attacked may go undetected due to its relative position with the monitoring node. Hence trust value calculated in such cases will be far from an accurate reflection of the true value.

Both the above cases will violate the assumption that μ's can viewed in ratio of observed matches and mismatches. For example in Table 5.2, we compare two trust evidences in scenario 1 and 2. According to the trust models discussed this far, both scenarios give trustworthiness of 1.00 for the optimistic trust heuristic, 0.97 for the beta distribution approach, meaning a high degree of trustworthiness. The trust scoring value $E^{j,i}$ discussed

thus far does not effectively capture the apparent uncertainty of nodes which do not have any mismatches, but have a high number of uncertain observations as opposed some node with lower uncertain observations. In such a case, we need to have a model that differentiates between scenario 1 and scenario 2 through a single trust value without waiting for convergence etc.

Table 5.2: Optimistic Model Example for high undecided; N=40

Scenario	α	β	μ	**Trust, Certainty**	**Beta Trust**
1	10	0	30	$1.00, 0.25$	0.97
2	31	0	9	$1.00, 0.775$	0.97
3	22	14	4	$0.61, 0.90$	0.60

The approaches discussed this far is not applicable for systems requiring a higher degree of reliability like mission critical systems. Binomial models are unable to distinguish in cases where there are high polarized ratings or highly neutral ratings [47], where there is high occurrence of one particular event which is neither positive or negative.

5.3 Dirichlet Expectation based Conservative Trust Model

Multinomial distribution is the generalization of the binomial distribution with $k > 2$ possible outcomes where each trial results in one of the k outcomes from a set of N possible trials. We can model match, mismatch and undecided as the possible outcomes on the inference over each channel; hence $k = 3$. Thus observation counts from the trust evidence fits very well with concept of multinomial distribution. Given this, observations data $\boldsymbol{D}$ for any node

can be treated as multinomial distribution given probabilities of occurrence of each outcome mathematically represented as $p(\boldsymbol{D}|\boldsymbol{X})$.

Here $\boldsymbol{D}$ is vector representing observed data such that $\boldsymbol{D} = \{d_1, d_2, d_3\}$; Similarly, $\boldsymbol{X}$ represents the probability vector denoting the probability of occurrence of each outcome such that $\boldsymbol{X} = \{x_1, x_2, x_3\}$. In general these are represented as $\boldsymbol{D} = \{d_i | 1 \leq i \leq k\}$ and $\boldsymbol{X} = \{x_i | 1 \leq i \leq k\}$. However initially $\boldsymbol{X}$ is unknown and hence the problem is estimation of parameters in $\boldsymbol{X}$ given evidence or data $\boldsymbol{D}$.

Dirichlet distribution effectively captures a sequence of outcomes where number of possible outcomes are more than 2 [45, 47]. The PDF of a Dirichlet distribution returns the subjective probabilities x_i (also known as degree of belief) for K_i number of rival events given each possible outcome i has been observed $d_i - 1$ times. Given that observation count $d_i - 1$ are known from the gathered evidence, we can calculate the probabilities of each event.

Dirichlet distribution is often used as a conjugate prior for a multinomial distribution, the unknown degree of belief associated with the three events can be calculated assuming prior and posterior preserve the same form [47]. In Bayesian systems, a prior probability distribution $p(x)$ is said to be conjugate to the class of distributions $p(D|x)$ if the resulting posterior $p(x|D)$ are in the same family as $p(x)$. In such a case, the resultant posterior $p(x|D)$ can be used as prior for further belief updates as incrementally new evidence $\boldsymbol{D}$ over time is received. For our work, the observation data counts form $\boldsymbol{D}$ and $\boldsymbol{X}$ is the probability parameter. In short, $\boldsymbol{X}$ is said to have a Dirichlet distribution with parameter $\boldsymbol{D}$ and is denoted as $\boldsymbol{X} \sim Dir(\boldsymbol{D})$.

Let us discuss the general theory of Dirichlet distribution. In terms of Bayesian systems, if $x_1, \cdots, x_i, \cdots x_k$ are the unknown probabilities associated with k events, and the evidence is d_i for event i, then the posterior degree of belief on each event i having accounted for evidence parameter d_i is given as $p(x_i|d_i) = \frac{p(d_i|x_i)p(x_i)}{p(d)}$. The evidence parameter d_i, is defined as $d_i = r_i + Ca_i$, where r_i represent the most recent count for event i and a_i represents a prior base rate and C represents an a-priori constant which dictates whether an informative or non-informative prior is assumed initially.

The posterior probability density function with variables $\vec{x} = (x_1, x_2, \cdots, x_k)$ and parameters $\vec{d} = (d_1, d_2, \cdots, d_k)$ is defined as

$$f(\vec{x}|\vec{d}) = \frac{\Gamma(\sum_{i=1}^{k} d_i)}{\prod_{i=1}^{k} \Gamma(d_i)} \prod_{i=1}^{k} x_i^{d_i - 1}, \tag{5.5}$$

where $x_1, x_2, \cdots, x_k > 0$, $\sum_{i=1}^{k} x_i = 1$, $d_1, \cdots, d_n > 0$. The relation between observation parameter d_i and actually observed outcomes r_i is that $r_i + C.a_i = d_i$, where $\sum_{i=1}^{k} a_i = 1$ and $C > 0, a_i > 0$ such that zero occurrence of an outcome preserves the condition that $d_i > 0$.

Modeling the trust given the assumptions will be equivalent to the expectation of the dirichlet distribution. We mathematically justify this later, with a formal mathematical proof in Section. 7.4. Hence, we denote the trust as given by the mean vector for Eqn. (5.5) and is given as

$$E(x_i|\vec{d}) = \frac{d_i}{\sum_{i=1}^{k} d_i} \tag{5.6}$$

where d_i is known as the total evidence count for event i. The degrees of belief associated with the outcomes are expressed as the mean of each outcome.

5.3.1 Applying Dirichlet model to trust evidence

We know $k = 3$, hence $\boldsymbol{D} = \{\vec{d}\} = \{d_1, d_2, d_3\}$ and $\boldsymbol{X} = \{\vec{x}\} = \{x_1, x_2, x_3\} : \sum_{i=1}^{k} a(x_i) = 1$. In our case, the most recent observation vector is the multinomial trust evidence $\boldsymbol{r} = \{\eta_\alpha, \eta_\beta, \eta_\mu\}$.

We observe $d_1 = \eta_\alpha + Ca(x_1)$, $d_2 = \eta_\beta + Ca(x_2)$ and $d_3 = \eta_\mu + Ca(x_3)$. Since there is no reason to believe a node has a particular pre-disposition to behave in a positive, negative or uncertain way, we assume a non-informative and hence a uniformly distributed prior. Since there are 3 outcomes, the prior initial base rate $a(x_i) = \frac{1}{3}$ and $C = 3$. In general, to preserve our assumption of uniformly distributed a priori, C would be equal to the cardinality of the state space. If we did not assume a uniformly distributed prior the value of C would have changed. Given this $d_1 = \eta_\alpha + 1$; $d_2 = \eta_\beta + 1$; $d_3 = \eta_\mu + 1$. Now that we have the parameters of the Dirichlet distribution, we can express the expected degrees of belief associated with the events of match, mismatch and undecided in terms of the observed trust evidence using Eqn. (5.6) as:

$$E_\alpha = \frac{\eta_\alpha + 1}{\eta_\alpha + 1 + \eta_\beta + 1 + \eta_\mu + 1} \tag{5.7}$$

Similarly, $E_\beta = \frac{\eta_\beta+1}{\eta_\alpha+\eta_\beta+\eta_\mu+3}$ and $E_\mu = \frac{\eta_\mu+1}{\eta_\alpha+\eta_\beta+\eta_\mu+3}$.

Hence for each node j, we have $E_\alpha = E^b_{ji}$ representing degree of belief, $E_\beta = E^d_{ji}$ representing degree of disbelief and $E_\mu = E^u_{ji}$ reflecting degree of uncertainty associated with behavior of node j based on gathered trust evidence of node i from the anomaly monitoring phase.

5.3.2 Interpreting belief as subjective logic for trust modeling

Subjective logic is directly comparable with a binary logic (trust or not to trust), probability calculus and classical probabilistic logic. Conclusions more correctly reflect actual scenario when there is ignorance and lack of information, and uncertainties that necessarily result from partial or uncertain input arguments [26, 45]. It is widely known as Josang's Belief Model. The proposition that a node will cooperate is either true or false and hence is a binary proposition. However, due to inherent uncertainty and imperfect knowledge caused by lack of evidence it is not possible to infer with certainty that the proposition is true or false. Hence we have only an *opinion* about this proposition and trust is often reported as the *expected opinion*. This translates the problem into degrees of belief, disbelief and uncertainty represented by E^b_{ji}, E^d_{ji} and E^u_{ji}, where $E^b_{ji} + E^d_{ji} + E^u_{ji} = 1$. Josang's belief model is hence used to deal with such data uncertainty in a proposition with binary state space, but with multinomial evidence [48] where one of the features express uncertainty. Josang's definition of opinion $\boldsymbol{\omega} = (b, d, u, a)$ is a quadruple where the components respectively correspond to

the belief, disbelief and uncertainty, and relative atomicity such that $b, d, u, a \in [0, 1]$ and $b + d + u = 1$. Then expected opinion pertinent to the positive interaction or belief is given as

$$E(\boldsymbol{\omega}) = b + au \tag{5.8}$$

where a is known as base rate or relative atomicity which determines how uncertainty contributes to the final expected opinion. Since the proposition that a node will cooperate or not is binary, we treat the value of a as 0.5 which is the value of relative atomicity in our model and $E^b_{ji} = b, E^d_{ji} = d, E^u_{ji} = u$. Hence we have expected opinion on the proposition that the node is cooperative or not is given by

$$E^{\boldsymbol{\omega}}_{ji} = E^b_{ji} + (a)E^u_{ji}. \tag{5.9}$$

5.3.3 An illustrative example of Dirichlet trust computation

The scenarios in Table 5.3, represent trust evidence on a particular time slot out of $N = 40$ channels for different nodes. Scenarios 1, 2, and 6 have occurrences of mismatches while 3, 4 and 5 do not. Intuitively, we would expect 3,4, and 5 to have higher trust than 1, 2 and 6. However, scenario 4 has high number of uncertain ratings as opposed to 5. The previously proposed Optimistic Trust Model (Model 1) cannot capture effect of relative uncertainty in one value. Hence $E^{j,i}$ gives the same answer for scenarios 4 and 5. *This ambiguity is resolved by the Dirichlet distribution based model.*

If we observed the corresponding values for the same in the next Model 2, given by $E^{\boldsymbol{\omega}}_{ji}$, we observe that Model 2 captures the presence of high number of uncertain ratings by generating a trust value of 0.61 for scenario 4, whereas giving a higher value of 0.86 to scenario 5, thus effectively differentiating between scenarios 4 and 5. We can also see that scenario 3 which has less uncertain ratings than 4 but more uncertain ratings than 5, has a trust value intermediate to the scenarios 4 and 5, thus preserving consistency in the rationale that given no evidence of mismatch, lower uncertainty should be awarded higher trust. Scenario 7, which has the most number of mismatches has a markedly low conservative trust value of 0.398. Hence the value $E^{\boldsymbol{\omega}}_{ji}$ does not have to necessarily depend on the assumption of uniformly random attacks or the non-zero probability of not detecting a single channel's attack.

Table 5.3: Expected Opinion and Conservative Trust; N=40

Scenario	α	β	μ	$E^{j,i}_{beta}$	$E^{\boldsymbol{\omega}}_{ji}$	w_{ji}
1	14	13	13	0.518	0.5116	0.045
2	19	18	3	0.513	0.5166	0.069
3	22	0	18	1.00	0.755	0.67
4	10	0	30	1.00	0.616	0.38
5	31	0	9	1.00	0.860	0.87
6	22	14	4	0.61	0.5929	0.31
7	14	22	4	0.38	0.398	−0.51

For Model 2 to work properly, we should expect that the trust of scenarios 1, 2 and 6 should be lower than that of scenarios 3, 4, and 5. We can verify that from the Table 5.3. However, among these scenarios where there is evidence of mismatches, scenario 6 has the least undecided and most number of matches compared to the other two. Hence scenario 6 achieves higher trust value than 1 and 2 but lower than scenario 4.

However, in future if scenario 4 gives a different result where evidence of mismatch is experienced then the values of it will change. In summary, we have showed how Model 2 effectively captures our concerns for a more conservative model which can express trustworthiness in a single value which captures not only highly polarized but also relative proportions of undecided ratings alongwith mismatches.

5.3.4 A Conservative trust weight metric

$E_{ji}^{\boldsymbol{\omega}}$ is a number between 0 and 1, which is a very narrow interval and makes the separation between non-trustworthy and the trustworthiness nodes difficult to depict or visualize. So we would like to make the trust values of the two classes linearly separable so that the malicious and honest nodes can be identified with a classification threshold with least false alarms and missed detections. For this, we intend map the trust values to a $E_{ji}^{\boldsymbol{\omega}}$ to a higher dimension to make it linearly separable popularly known in machine learning as a *kernel trick*. We use the Shapley log scaling equation to transform $E_{ji}^{\boldsymbol{\omega}}$ to a generic value on the real line where non-trustworthy nodes have a monotonically decreasing value and trustworthy nodes have monotonically increasing weights. We then report the final normalized weight by giving a value between [-1,1]. The Shapley value based weight is given as

$$r_{E_{ji}^{\boldsymbol{\omega}}} = log_2 \left(\frac{E_{ji}^{\boldsymbol{\omega}}}{1 - E_{ji}^{\boldsymbol{\omega}}} \right) \tag{5.10}$$

The normalized conservative trust weight is given by

$$w_{ji} = \begin{cases} 1 - e^{-|r_{E_{ji}^{\boldsymbol{\omega}}}|} & \text{if } r_{E_{ji}{}^{\boldsymbol{\omega}}} > 0; \\ -(1 - e^{-|r_{E_{ji}^{\omega}}|}) & \text{if } r_{E_{ji}{}^{\omega}} < 0; \\ 0 & \text{if } r_{E_{ji}{}^{\omega}} = 0 \end{cases} \tag{5.11}$$

where $w_{ji} \in [-1, 1]$. For quantifying reputation of a node, performance of node isolation and plot purposes, we mostly use the absolute value of node j's final trust rating which is the average of all trust ratings w_{ji} calculated by node j's neighbors. Hence *average trust rating* or the *reputation* of a particular node j can be represented as w_j. All w_{ji} corresponding the scenarios are discussed in Table 5.3. The trends are same as $E_{ji}^{\boldsymbol{\omega}}$, but the separation between the nodes are large and distinct and that will help separating nodes into malicious or honest with a high degree of certainty as we will show in Chapter 6.

CHAPTER 6: APPLICATION OF TRUST MODELS

In this chapter, we discuss how the proposed trust models can be applied for different operational decisions to preserve the integrity and security of the network in presence of malicious nodes employing different intelligent strategies. First, we provide a discussion on how to use trust models for identification and isolation of malicious nodes with a high degree of certainty using a machine learning approach. Then we propose an asymmetric weighted moving average scheme for trust maintenance and management under on-off SSDF attacks. Then we discuss the use of optimistic and conservative trust models for robust information fusion at a node. Finally, we discuss inversion based fusion schema- an inclusive approach which uses observed invert sequence to utilize misleading information of malicious nodes for the benefit of the network.

6.1 Trust based Malicious Node Identification: A Machine Learning approach

Reputation or trust based systems can be used to identify malicious nodes if there is sufficient time to observe such nodes. Let at any time, the conservative trust of node j calculated by a particular neighbor i be represented as $w_{ji}(t)$. The steady state value can be expressed as a moving average by w_{ji}. For malicious node identification, we treat the reputation of node

j as the *average* of steady state conservative trust values as seen by all its neighbors. This average is usually done by a policy enforcement entity, who collects the trust weights of node j from all its neighbors. We denote this average as w_j dropping the suffix i. The particular implementation of how the enforcement entity collects of the individual trust ratings are out of the scope of this work. We treat w_j as the long term reputation of a node j which is used for malicious node identification.

6.1.1 Need for a learning approach

We need a learning approach because a decision as drastic as declaring a node malicious, isolating a node from the network or reporting policy violation, we have to make sure the model is general enough and is able to classify for a wide range of scenarios. We seek to propose a framework that is learned enough to work irrespective of the type of pathloss environment, topology, and magnitudes of attack. We do so using a machine learning approach, whereby we simulate training data sets under various values of P_{attack} and under different pathloss environments. Our objective is to find an optimal threshold weight $w_{classify}$ that can decide whether a node is malicious or not. We use the concept of support vector machines (SVM) to identify $w_{classify}$.

Pathloss environment plays a key role in degree of undecided: Apart from spatial orientation, the pathloss environment plays a key role in determining the inherent uncertainty induced by

the proposed monitoring technique. In general, for all practical terrains where obstructions are present, pathloss factors vary from 2.8 to 6 [39].

Through our experiments we see that a network with pathloss factor of 4 (neither high or low) induces maximum uncertainty, while pathloss factors lower or higher than 4 (say 3 or 5), lowers the average degree of undecided. One possible explanation for this is that since signals decay too fast or too slow for higher and lower pathloss factors, the chances of both the P_{high} and P_{low} being either above or below the particular normalizing threshold value γ_{th} within the radius s_{ij} is increased. Given that, all the other factors remain same, if the pathloss factor is neither too high or too low, the chances of both P_{high} and P_{low} being above or below γ_{th} *decreases*. This is evident from our experiments which calculates the average degree of uncertainty for all nodes across the network for different pathloss environments listed in Table. 6.1.

Table 6.1: Effect of pathloss on uncertainty; Worst case $P_{attack} = 0.50$

Pathloss	**Average** E_μ
3	0.166065
4	0.426087
5	0.305750
5.5	0.200618

While pathloss factor as low as 3 or as high as 5.5, generates lower average E_μ of 0.16 and 0.20 respectively, an intermediate pathloss factor of 4, produces E_μ as high as 0.42. Lack of information increases the difficulty of a classification problem. Hence we conclude that classification becomes harder when pathloss is around 4 and easier when the pathloss factors are on the extremes. Hence we motivate the need for learning over training sets

considering different pathloss environments, so that we can find an optimal classification threshold, that is able to generalize effective classification over a testing set considering a variety of environments.

Trust or reputation is dependent on magnitude of attack: Our proposed mechanism will induce trust and reputation values that are directly dependent on the magnitude of attack; viz. greater the magnitude lesser the trust. An example of this is shown in Fig. 6.1. This is intuitive since under an effective monitoring mechanism, the more a node attacks the more it exposes itself and it becomes easy to detect anomalies. However, a malicious node should have some minimum magnitude of attack that should justify its malicious behavior. What will be the minimum magnitude of attack for an adversary is an open ended question and depends on a number of factors. Whatever be the magnitude of attack, our model should be able to detect such nodes with a high degree of certainty. In general, if we can detect for lower magnitudes of attack, we can automatically detect for higher magnitudes of attack as well. Hence lower magnitudes of attack are a limiting factor.

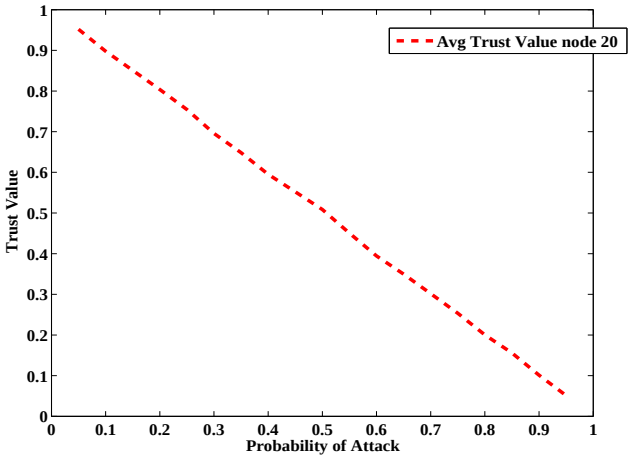
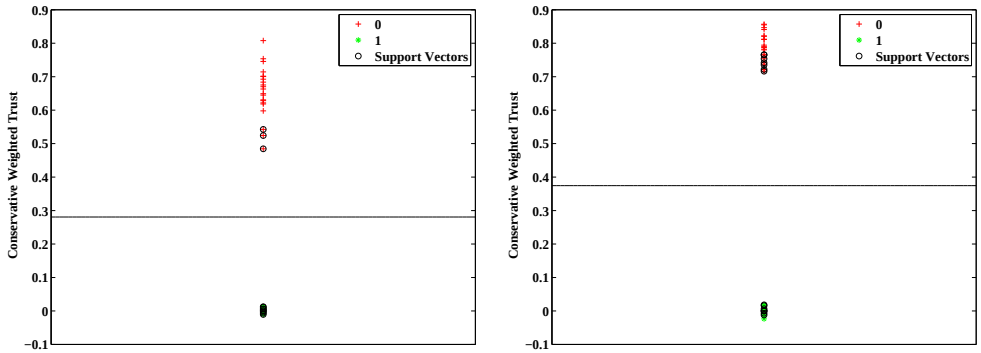


Figure 6.1: Relationship between Trust and Weight

6.1.2 Choice of training data sets

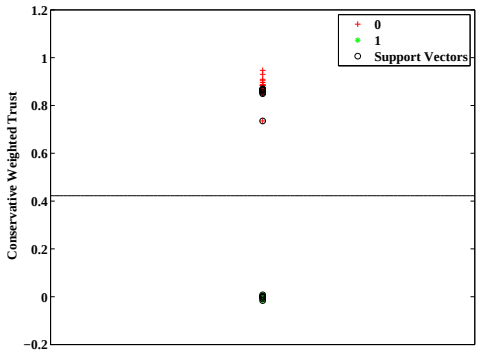
We use one training set for each pathloss $\omega = 3,\ 4$ and 5. For each training set we choose magnitude of attack as 0.5. We initially train with 30 nodes randomly distributed over an area of 100X100 units, with 30 units of sharing radius. 9 nodes are programmed malicious and all of which have the same magnitude of attack. We observe the trust values of the honest and malicious nodes. We run a support vector machine (SVM) over training examples which maps the trust values into support vectors and finds the optimal hyperplane which in our case is a single line due to the linear nature of the data with only one feature i.e., the trust value. In Figure 6.2(a), + represents labels corresponding honest nodes and $*$ represents labels corresponding to malicious node. The optimal hyperplane is a line which is the output of the SVM, that maximizes the margin between the support vectors corresponding to the closest honest and the closest malicious node over the feature space. The optimal hyperplane for each training set represents a candidate threshold that may be chosen to classify whether a node is malicious or not. We represent our training sets as $T_l = T(\omega, P_{attack})$ where l represents the l^{th} training set. Hence $T_1 = T(3, 0.5)$ is the first training set with pathloss factor of 3 and magnitude of attack as 0.5 with the attack measure as P_{attack}. Similarly, $T_2 = T(4, 0.5)$ and $T_3 = T(5, 0.5)$. Training results on SVM for $T_1,\ T_2,\ T_3$ are shown in Figures. 6.2(a), 6.2(b), 6.2(c).

We also find the effect of varying P_{attack} in Fig. 6.3(a) and Fig. 6.3(b) with $P_{attack} = 0.5$ and $P_{attack} = 0.8$ for network with $\omega = 3$ and fourth training example $T_4 = T(3, 0.8)$.



(a)

(b)



(c)

Figure 6.2: (a) Environment with Pathloss=4; $P_{attack} = 0.50$ (b) Environment with Pathloss=5; $P_{attack} = 0.50$(c) Environment with Pathloss=3; $P_{attack} = 0.50$

6.1.3 Rationale for model selection

The lower region of the SVM result contains support vectors with labels corresponding to malicious nodes and the upper region contains labels that correspond to the honest ones. Now we run a SVM for various environments for $P_{attack} = 0.5$. Choosing of thresholds can be tunable and adjusted according to the requirements of the system. Here we aim at identifying the training set mimicking the worst case scenario for classification. We do

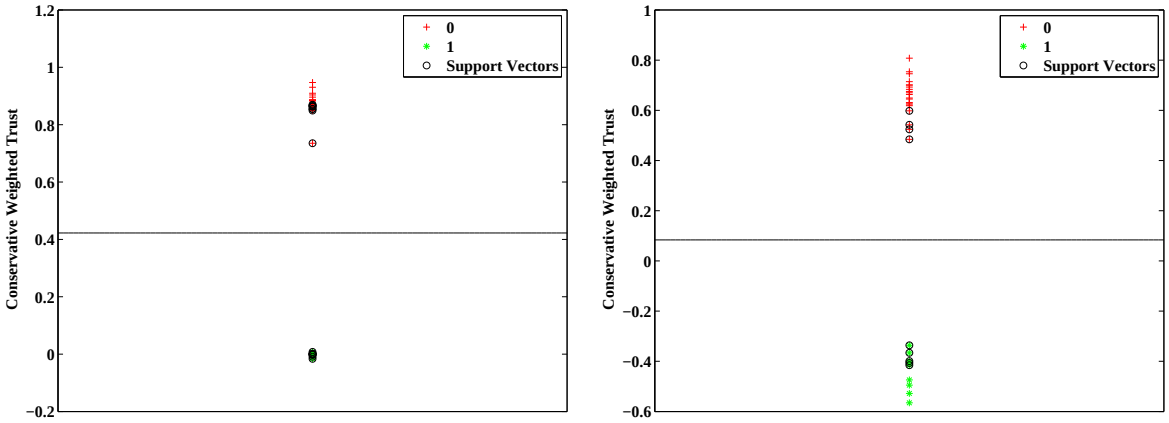
not show all the higher probabilities of attack because the classification gets easier as nodes become more aggressive. If we isolate for a lower magnitude of attack, automatic isolation for higher magnitude follows.

In Figs. 6.2(a), 6.2(b), 6.2(c), the horizontal line separating the trust domain into two regions can be considered as the candidate threshold for the relevant testing set. For designing threshold, we have the following candidate thresholds: $(0.42, 0.29, 0.39)$. We have two alternatives:

Case 1: A single common threshold: If we wish to design a pessimistic system, with higher reliability that works regardless all environments and for different magnitudes of attack, we should choose the training set that represents the worst case in terms of uncertainty in the trust evidence. Hence we select training set T_2 (known as model selection), which has a pathloss factor of 4 and a $P_{attack} = 0.5$. The optimal hyperplane that separates the training examples of T_2 is approximately equal to 0.29. Hence we can treat, $w_{classify} = 0.29$, for classification of malicious nodes.

Case 2: Different threshold based on pathloss environment: Alternatively, we can have different thresholds for each pathloss environment. Then different thresholds can be designed depending on the relevant environment. We can take 0.42 for pathloss 3, 0.29, for pathloss 4, 0.39 for pathloss 5.

We explore both the above options using simulations and performance modeling, and report which alternative is better.



(a) (b)

Figure 6.3: (a) Environment with Pathloss=3; $P_{attack} = 0.50$ (b) Environment with Pathloss=3; $P_{attack} = 0.80$

6.1.4 Choice of testing data sets

For the testing data sets, we assume P_{attack} is *not same* for all malicious nodes. We have a network with 100 nodes where nodes are randomly chosen to be malicious and such nodes are divided into three groups having a low, medium and high P_{attack} values of 0.3, 0.5 and 0.8 respectively. The threshold selected from the chosen model in our training should be able to capture such nodes, under any pathloss factor. The testing set is also a bigger network of 600x600 area with a sharing radius of 200 units. We take results of networks under different pathloss factors 3.2, 4 and 4.8. We show the performance of the chosen classification model over these testing sets in the simulation section. Also for most other simulation results, we keep the pathloss factor of 4, as this represents the worst case from the point of lack of evidence.

6.1.5 Trust updates over time

For our given attack model (except On-Off attacks), a cumulative moving average also known as equally weighted moving average for maintaining node reputation makes sense because in order to isolate a node we need to keep its long term history of behavior. As instantaneous trust value of node j as calculated by node its neighbor i is calculated on time t as $w_{ji}(t)$. The cumulative moving average is the average trust at t for all of the interactions up to that point using the previous cumulative average. Hence at any time t, a node's long term average trust, $w_{ji}^{mavg}(t)$ is updated as

$$w_{ji}^{mavg}(t) = \frac{(t-1)w_{ji}^{mavg}(t-1) + w_{ji}(t)}{t}$$

The cumulative moving average is essential to characterize long term behavior or strategies of a node because it does not cause loss of information over time unlike exponential weighted moving average. The reputation of node j used to decide whether node j is malicious or not, is average of all $w_{ji}^{mavg}(t)$ pairs for each neighbor i who receives node j's spectrum data.

6.2 An asymmetric trust update scheme for On-Off Attacks: Special Case

Till now we have discussed attackers who may either be resource constrained or might have a particular magnitude of attack (probabilistic or not). In such attacks, there is no preference on what time slots attacks will be launched. In such cases, trust values over time can be

updated as equally weighted moving average, that would reflect the true behavior over time. However, in on-off attacks, nodes have preferences over time periods where a node may choose not to attack for some time and then attack for some time with a random magnitude. In such a case, both equally weighted moving average or exponentially weighted moving average would not reflect true behavior of the node. An equally weighted moving average will *lag* in reflecting such attacks, while weighted moving averages will enable a malicious node to *quickly recover or redeem its reputation.* In such cases, the trust management framework should be such that a node with a history of malicious or anomalous behavior *should not* be allowed to recover its trust value quickly even though it starts behaving well after a short burst of attack. In Fig. 6.4, we show an example of reputation of node 20 as seen by node 29. Malicious node 20 employs an on-off attack where it divides the time domain of 500 slots in five stages; 'Stage 1' ranging from $t = 0 - 100$, 'Stage 2' ranging from $t = 101 - 150$, 'Stage 3' ranging from $t = 151 - 250$, 'Stage 4' ranging from $t = 251 - 300$ and finally 'Stage 5' ranging from $t = 301 - 500$. In the 'Stage 1' of 100 slots, malicious node 20 does not attack in a bid to gain a high trust value initially. In Stage 2, it attacks for the next 50 time slots with a random magnitude on each of the time slots. In Stage 3, it does not attack on any of the 100 slots. In Stage 4, it again attacks on 50 slots with a random attack magnitude. In Stage 5, it again behaves cooperatively. Imagine a node isolation algorithm that checks for node trust values every 50 slots. Say there is a system which uses a threshold of zero below which nodes should be identified as malicious. On 100-th, 200-th, 250-th, 350-th, 400-th, 450-th and 500-th slot, the node is deemed *not malicious* by both trust update schemes although we know

the node is employing a stealthy on-off attack. We see that equally weighted or cumulative average reacts too slow and fails to reflect malicious nature even at the end of the Stage 2. On the other hand, exponential weighted moving average although detects attacks quickly also allow such nodes to quickly recover their reputation on 151-th and 301-th slot. Hence there is need for a special trust update scheme that would restrict the average reputation to improve quickly even when a node starts behaving cooperatively after a period of malicious activity. At the same time it should also be responsive enough to quickly decrease reputation when a node after building high reputation starts acting maliciously.

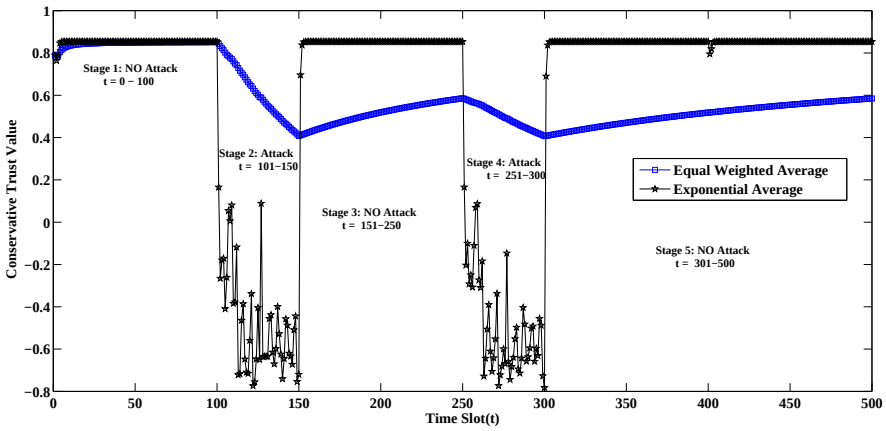


Figure 6.4: Problems of weighted moving averages under on-off attacks

We propose a technique to deal with on-off attacks from a socially inspired concept that bad actions are far more remembered than our good actions. This forms the basis of our asymmetric weighted moving average scheme, where slots with instantaneous trust values $w_{ji}(t)$ lower than a threshold Γ_{on-off} are given more weight than time slots where $w_{ji}(t)$ has higher values. The value of Γ_{on-off} is dictated by a system specific risk attitude and

defines what can termed as sufficiently good behavior. In the update of trust values, there are two important things; the cumulative average and current trust value. We introduce four weighting factors χ_a, $\chi_{b_{max}}$, $\chi_{c_{min}}$ and χ_d such that $0 < \chi_a < 1$; $0 << \chi_{b_{max}} < 1$; $0 < \chi_{c_{min}} << 1$ and $0 < \chi_d < 1$. Note the fact that $\chi_{c_{min}}$ is much much less than $\chi_{b_{max}}$ introduces an asymmetry. Now there may be four possible scenarios at time t with regard to On-Off attacks.

Case(a): $w_{ji}^{mavg}(t-1) > \Gamma_{on-off}$ and $w_{ji}(t) > \Gamma_{on-off}$

Case(b): $w_{ji}^{mavg}(t-1) > \Gamma_{on-off}$ and $w_{ji}(t) \leq \Gamma_{on-off}$

Case(c): $w_{ji}^{mavg}(t-1) \leq \Gamma_{on-off}$ and $w_{ji}(t) > \Gamma_{on-off}$

Case(d): $w_{ji}^{mavg}(t-1) \leq \Gamma_{on-off}$ and $w_{ji}(t) \leq \Gamma_{on-off}$

In Case (a), a cumulative average higher than Γ_{on-off} suggests a node is maintaining a sufficiently good behavior. If the current trust value is also higher than Γ_{on-off} then it suggests continuity of the good behavior. Hence continuing good behavior is rewarded with a high weighting factor χ_a to $w_{ji}(t)$ and low weightage given to $w_{ji}^{mavg}(t-1)$ using $1-\chi_a$. We name χ_a as a *rewarding factor* such that $1 > \chi_a > 0$. It helps a historically good node to improve or atleast maintain its reputation if it also has behaved in a cooperative manner in this time slot t. Hence for Case (a) cumulative trust is updated as:

$$w_{ji}^{mavg}(t) = (1-\chi_a) \times w_{ji}^{mavg}(t-1) + \chi_a \times w_{ji}(t)$$

In Case (b), a cumulative average higher than Γ_{on-off} and $w_{ji}(t) \leq \Gamma_{on-off}$ suggests a node maintaining a sufficiently good behavior upto time $t-1$ and then initiated some anomalous behavior. Hence all the good behavior until now needs to be forgotten and very high weight be given to current slot's anomalous behavior. This will cause the node's cumulative trust value to quickly decrease. Once this happens Case(c) would ensure that the cumulative trust is not able to redeem itself quickly. Hence $w_{ji}(t)$ is weighted with a high value $\chi_{b_{max}}$ such that $1 > \chi_{b_{max}} >> 0$ and $w_{ji}^{mavg(t-1)}$ is weighted using $1 - \chi_{b_{max}}$. We name $\chi_{b_{max}}$ as *punishment factor*. The higher the value of punishment factor the quicker and more severe the system will be to new evidence of malicious behavior. In such a case cumulative trust is updated as:

$$w_{ji}^{mavg}(t) = (1 - \chi_{b_{max}}) \times w_{ji}^{mavg}(t-1) + \chi_{b_{max}} \times w_{ji}(t)$$

In Case(c), a cumulative average lower than Γ_{on-off} but a current trust value $w_{ji}(t)$ higher than Γ_{on-off} signifies a node whose current behavior is cooperative but has a history of anomalous behavior which may be as recent as $t-1$. Hence even though $w_{ji}(t)$ may be high we assign it a very low weight $\chi_{c_{min}}$ such that $0 < \chi_{c_{min}} << 1$ and assign $1 - \chi_{c_{min}}$ to $w_{ji}^{mavg(t-1)}$. We name $\chi_{c_{min}}$ as the *redemption factor* that controls how fast or slow a node with malicious history can redeem its trustworthiness if it shows good behavior for a sufficiently long time. Redemption factors also make it possible for nodes who experienced noise redeem their trust values. A low redemption factor ensures that trust value is not

increased quickly even though a node starts to behave honestly after a period of malicious behavior. In this case cumulative trust is updated as:

$$w_{ji}^{mavg}(t) = (1 - \chi_{c_{min}}) \times w_{ji}^{mavg}(t-1) + \chi_{c_{min}} \times w_{ji}(t)$$

In Case(d), both cumulative average and current trust value of node j are below Γ_{on-off} indicating continuing anomalous behavior. In such a case, we provide χ_d known as *retrogression factor* as weight to the current value and $1 - \chi_d$ weight to cumulative average such that trust is updated as:

$$w_{ji}^{mavg}(t) = (1 - \chi_d) \times w_{ji}^{mavg}(t-1) + \chi_d \times w_{ji}(t)$$

.

The above scheme termed as asymmetric weighted moving average is effective in defending against On-Off attacks which is not possible using equally weighted or exponential weighted moving averages. In simulation section, we also show that this can also be effective to distinguish malicious nodes and nodes experiencing intermittent noise. In simulation section we show the performance with various values of $\chi_{c_{min}}$, $\chi_{b_{max}}$, χ_a and χ_d.

6.3 Trust based Robust Information Fusion: Isolation Approach

In this section, we show how trust values can be used to filter out anomalous reports from being considered while a node fuses reports to obtain the global spectrum occupancy inference. We propose an optimistic Trust based fusion and conservative trust based fusion for different systems to filter out anomalous reports on each of the time slots. This enhances the robustness of the fused spectrum occupancy information at nodes.

6.3.1 Optimistic trust based fusion: Beta distribution model

Though steady state values of trust is ideal for node identification and classification, such values on the other hand do not necessarily contribute towards robust fusion under adversarial conditions. In dynamic systems, *waiting for convergence* for filtering out spurious reports is not an option. This is relevant in an ad-hoc CR network in three ways. First, nodes may be mobile, hence neither the neighbors of a node nor nodes themselves are long lasting entities. At the same time however, a decision on the channel occupancy needs to be made from the reports of current neighbors. Hence maintaining history of updates of trust value of neighbors may not be a prudent idea in a distributed CR network. Second, we seek to investigate whether performance of the system can be improved while being in a transient state before convergence of trust value is attained. Third, the attack measure P_{attack} is a long term average value, but a particular realization of P_{attack} at a particular time, may be

different from the mean value of P_{attack}. For. e.g., a malicious node has $P_{attack} = 0.60$, but at a particular time slot only 0.3 fraction of channel may have been attacked. Thus, it has contributed on 70% of the channels on that time slot. In such a case, if we isolate this report based on long term reputation based exclusion, we will lose the majority of honest opinions alongwith the minority falsified opinions. However, instantaneous or transient trust or reputation is an index of behavior on the current interaction. Since only current interactions are important as far as utility of spectrum sensing usage reports are concerned, steady state trust values should not be used as a metric for robust fusion based on exclusion of spectrum sensing reports on each time slot.

Using the computed *instantaneous trust* coefficients, we study the performance of two fusion schemes: blind fusion and trust-based filtered fusion.

6.3.1.1 Blind fusion

Blind fusion is fusion when no defense mechanism is employed and will be used for comparison purpose. A node considers all reports it receives from its neighbors. For blind fusion, any node i considers all its neighbors to be honest and includes B^j_{adv} from all its neighbors along with its own B^i_{act}. We formally define Blind Fusion as $BF^i_{blind} = \nabla[B^j_{adv} \oplus B^i_{act}]$, $j \in N_i$ where ∇ is the operator for majority voting rule. Majority voting is a popular fusion rule where final fused inference on a channel is based on what at least half the neighboring nodes advertise with all the nodes treated equally. $\oplus$ is the operator for combination.

6.3.1.2 Optimistic trust-based fusion: Beta expectation model

We propose a fusion scheme whereby we only consider neighboring nodes whose $E_t^{j,i}$ is higher than some trust threshold, Γ_{opt}. (Later in Chapter 8, we show how to find the optimal threshold). Thus, for trust-based fusion, node i only considers those neighbors whose $E_t^{j,i} \geq \Gamma_{opt}$. In effect, the fusion is done with information from trusted nodes only.

$$\text{If } E_t^{j,i} \begin{cases} \geq \Gamma_{opt} & \text{Node } j\text{'s report is Trusted} \\ < \Gamma_{opt} & \text{Node } j\text{'s report not trusted} \end{cases} \tag{6.1}$$

We define optimistic Trust based fusion result as; $TBF^i = \nabla[TFS_i \oplus B^i_{act}]$, where TFS_i is the trusted fusion set of binary vectors accumulated by node i using Eqn.(6.1), which includes B^j_{adv} of trusted nodes only.

Although the nodes are not aware of the ideal scenario, we are aware of what would have been the ideal fusion result, which is the case when for all node $j \in N_i, B^j_{act} = B^j_{adv}$, so we define Ideal fusion result for node i as $BF^i_{ideal} = \nabla[B^j_{act} \oplus B^j_{act}]$. This is later used for comparing the performance of fusion by measuring deviation from the ideal result.

6.3.1.3 Conservative trust based fusion: Conservative trust model

Similar to the above scheme, we propose a fusion scheme with a conservative threshold Γ^c_{opt}, for the conservative system. (Later in Chapter 8, we show how to find the optimal threshold).

Thus, for trust-based fusion, node i only considers those neighbors whose $w_t^{j,i} \geq \Gamma_{opt}^c$. In effect, the fusion is done with information from trusted nodes only.

$$\text{If } w_t^{j,i} \begin{cases} \geq \Gamma_{opt}^c & \text{Node } j\text{'s report is Trusted} \\ < \Gamma_{opt}^c & \text{Node } j\text{'s report not trusted} \end{cases} \tag{6.2}$$

We define Conservative Trust based Fusion result as $CTBF^i = \nabla[CTFS_i \oplus B_{act}^i]$, where $CTFS_i$ is the conservative trusted fusion set of binary vectors accumulated by node i using Eqn.(6.2), which includes B_{adv}^j of trusted nodes only.

6.4 Inversion based fusion schemas: An inclusive approach

We introduce a new approach that utilizes the observed invert sequence discussed in the anomaly monitoring phase in Section 4.4. Using invert sequence $IS^{j,i}$ of a particular neighbor node, we propose a framework whereby instead of disregarding falsified information from potential malicious nodes, we utilize the false information to our advantage. We calculate the trust of each node in the same way as discussed in the optimistic beta expectation model. Once the trust values are known, we use a log-weighted metric to distinguish the malicious nodes from others. Then, we use weighted threshold based *Selective Inversion (SI)* fusion and *Complete Inversion (CI)* fusion schemes to effectively fuse data obtained from *all*

nodes. We also propose a combination of the two inversion schemes which provides better performance for all probabilities of attack. We find the conditions for which the combination works better. The fraction of mismatches (false alarms and missed detections), in presence of malicious nodes for different probabilities of attack and node densities are considered as a performance metric. We are able to show that fraction of mismatches for the proposed fusion techniques is always less than the optimistic trust based fusion which disregards malicious nodes in fusion. At the end, we demonstrate that we could utilize false information sent by malicious nodes to increase the gain in cooperative spectrum sensing. Our objective is to intelligently invert elements of the occupancy vector that are sent my malicious nodes.

6.4.1 A log-weighted metric based on trust

First, we need to figure out how much weight do we give to each node. Note, due to the variability in the trust values, we cannot treat all nodes equally. If trust of node j as computed by node i is $E_{beta}^{j,i}$, we denote its corresponding weight as:

$$W_i^j = log_e[\frac{E_{beta}^{j,i}}{1 - E_{beta}^{j,i}}] \tag{6.3}$$

Although we expect trustworthy nodes to have higher trusts, it is not possible to have a clear separation between honest and malicious nodes if use trust metrics which are between 0 and 1, as the range is too small. Hence we use the above equation to clearly

distinguish between two classes of nodes. The above equation ensures negative weights for nodes whose trusts values are below 0.5 and positive otherwise as shown in Fig. 6.5. This ensures that the weights are monotonically increasing for honest nodes and monotonically decreasing for malicious nodes. With this equation a small decrease in trust for a small increase in malicious behavior will ensure a large decrease in weight of that node. Then we use weights to decide the criterion for inversion.

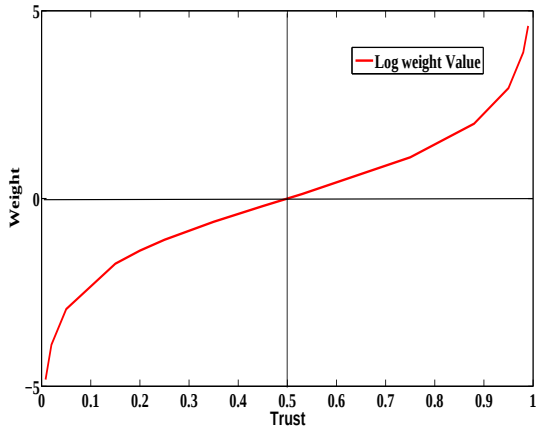


Figure 6.5: Relation between Beta trust and log weighted metric

6.4.2 Criterion for inversion based fusion

We define two thresholds: minimum (W_{min}) and optimal (W_{opt}). Nodes for which the weights are more than W_{opt} are considered honest and no change is made to their advertised vectors. Nodes for which the weights are less than W_{min} are deemed totally malicious; thus every element in their advertised vectors is inverted. However, the elements in the advertised vectors are selectively inverted (bits flipped) for the nodes whose weights lie between W_{min}

and W_{opt} using the invert sequence. That is,

$$\text{If } W_i^j \begin{cases} > W_{opt} & \text{No Inversion} \\ \leq W_{min} & \text{Complete Inversion} \\ \leq W_{opt} \text{ and } \geq W_{min} & \text{Selective Inversion} \end{cases} \tag{6.4}$$

The challenge lies in determination of these thresholds– which will be determined through simulations. We symbolically represent inversion based fusion IBF, which is a combination of Complete Inversion (CI) and Selective Inversion (SI).

6.4.2.1 Complete Inversion (CI)

For all malicious neighbors of i (identified by Eqn. (6.1)), we invert all elements of the advertised vector. In this case, the expected number of channels on which we get the correct opinion is proportional to P_{attack}. For example, if a malicious node modifies 80% of the observed data, we get back the actual sensed opinion on 80% of the channels after inversion. Of course, the hind side of complete inversion is that the correct information becomes incorrect. Thus it works better for higher P_{attack} values. Hence it is used when weights are very low (i.e., lower than W_{min}). We prove this claim in the simulation section.

6.4.2.2 Selective Inversion (SI)

For all neighbors i, we get the invert sequence IS_i^j from the Eqn. (4.10). IS_i^j indicates all channels with mismatches. For neighbors whose weights lie between W_{opt} and W_{min}, we seek to selectively invert the advertised values in B_{adv}^j for channels comprising the set IS_i^j. Such inversion forms a new vector for all non-trusted neighbors. This scheme is applicable for lower P_{attack}.

The reason for using a combination of two inversion based fusion schemes is because the network does not know P_{attack}. However, the trust weights depend on P_{attack} and as trust changes, the type of inversion scheme that works better is employed. In simulation section, how this is implemented.

6.5 Performance analysis measures

In this section, we provide the performance analysis measures which are used to decide whether proposed malicious node identification and robust fusion models work properly to provide operational assurance. For the malicious node identification, we use the percentage of true negative detection as a performance measure. The higher the percentage the better is the performance. For the performance analysis of robust fusion we use percentage of mismatches from ideal fusion result. Although the nodes are not aware of the ideal scenario,

we are aware of what would have been the ideal fusion result, which is the case when for all node $j \in N_i, B^j_{act} = B^j_{adv}$. So, we define $BF^i_{ideal} = \nabla[B^j_{act} \oplus B^j_{act}]$. This ideal fusion result is used for comparing the performance of proposed fusion techniques by *measuring the deviation* (fraction or percentage of mismatches) of the concerned fusion technique from the ideal result. The lesser this deviation, the better is the performance of the scheme. We use percentage of mismatches from the ideal result as a performance metric, to compare the degrees of improvement in cooperation sensing result. We also compare proposed schemes with a scenario where there is no defense mechanism known as $BF^i_{blind} = \nabla[B^j_{adv} \oplus B^i_{act}]$ which models a situation of coexistence with malicious nodes without any defensive scheme. We show heavy improvement of the proposed schemes compared to blind fusion in the simulation. In addition to that we show that the inclusive approach of inversion based fusion scheme works better than the exclusionary trust based fusion which disregards fusion reports.

6.6 Summary

In this chapter, we proposed a reliable malicious node detection technique using machine learning approach. We proposed a technique that utilizes misleading information sent by malicious nodes for the purpose of cooperative sensing in cognitive radio networks. Contrary to common approaches, where information sent by malicious nodes are simply excluded for any decision making, we follow inclusive approaches to exploit even the misleading information. We argue that if the trustworthiness of each malicious node can be computed, then we

can appropriately negate the false information. To this end, we use a log-weighted function to compute the trust value of every node. We present two schemes (selective inversion and complete inversion) for inverting the occupancy information of the channels. The combination of these two inversion schemes is used which yields better spectrum occupancy estimates than trust-based and blind fusion schemes for all probabilities of attack. We also find the conditions for which one scheme works better than the other.

CHAPTER 7: BAYESIAN INFERENCE BASED CHANNEL PREFERENCE UNDER COLLABORATIVE SELFISH SSDF ATTACKS

The previous chapters have discussed SSDF attacks that are not coordinated and the rationale of dishonest nodes was assumed to be only malicious. However, SSDF attacks are more severe when a collaborative SSDF attack is launched by a coalition of *selfish nodes* on select channels. Nodes with a selfish rationale usually have some preferred channels that they seek to obtain; hence they falsify only on those channels. In such cases, magnitude of attack is much low compared to malicious SSDF attacks. Meanwhile, defense against such collaborative attacks is difficult with popularly used voting based inference models as acknowledged in [21, 36, 49]. This chapter introduces a *channel centric approach* based on Bayesian inference that indicates how much the collective decision on a channel can be trusted. Using our anomaly monitoring technique, we check if the reports sent by a node match with the expected occupancy and classify the outcomes into three categories: i) if there is a match, ii) if there is a mismatch, and iii) if it cannot be decided. Based on the measured observations over time from a channel centric approach, we estimate the parameters of the hypothesis of match and mismatch events using a multinomial Bayesian based inference. We quantitatively define the trust as the difference between the posterior beliefs associated with matches and that of mismatches. The posterior beliefs are updated based on a weighted average of

the prior information on the belief itself and the recently observed data. We also show, how can we distinguish between selfish and malicious nodes. We conduct simulation experiments that show that the proposed trust model is able to distinguish the attacked channels from the non-attacked ones. Also, a node is able to rank the channels based on how trustworthy the inference on a channel is. We are also able to show that attacked channels have significantly lower trust values than channels that are not.

7.1 Motivation of a Channel Centric Bayesian Trust Framework

To counter SSDF attacks, the common approach has been on the *identification* or *isolation* of dishonest nodes [8, 36]. But, there has not been much effort on distinguishing between selfish and malicious rationale of rogue nodes, or the effect of a coalition formation. While a malicious attacker's only objective is maximum damage to other nodes in a network, selfish attacks are launched by rational nodes on a few strategic channels so as to gain some tangible benefit. A classic example of a collaborative selfish SSDF attack is a group of nodes belonging to a certain network provider which wants access to some selected channel(s); hence all nodes belonging to that provider send false information for those channels while truthfully reporting on all other channels in order to remain undetected. Compared to the number of malicious nodes, the number of selfish nodes can be high, considering every node has a natural proclivity to maximize its benefits. Hence selfish attacks are more plausible form of a vulnerability. But total isolation of all such selfish nodes is perhaps not the right

approach. This is because in such a case the fusion process *loses the correct information* that the selfish nodes might share [9]. This loss is even more pronounced when the selfish nodes collaboratively falsify (on same channel sets) with a low probability and tell the truth more often [7]. In fact, it is often difficult to detect such selfish behavior of nodes through *node centric* malicious detection techniques. Thus, instead of isolating and not considering the selfish nodes at all, we ought to consider shared information on specific channels and discount or completely ignore information on other channels. Thus, we argue for the need of a *channel-centric defense* instead of a node-centric defense.

7.2 System Model for Selfish SSDF Attacks

We consider a distributed secondary cognitive radio network with honest nodes denoted by H and a set of selfish nodes denoted by S. We consider node i, running our proposed scheme. We consider N as the number of neighbors of node i, at any point of time. Each node i fuses the spectrum sensing data it receives from its set of neighboring nodes, denoted by N^i. For any channel k, there are N_k^i advertised opinions on the occupancy of channel k at node i and K is the total number of channels being monitored. We use the same model for spectrum sensing and reporting as was discussed in Section 4.1. The only difference is S denote set of selfish nodes, K denote total number of channels, and N_k denote the total number of opinions received on any channel k. We consider that the selfish nodes do not report their occupancy vectors truthfully; rather they inject errors in their occupancy vectors by flipping

the bits in the vector collaboratively. The selfish nodes collaborate to attack (i.e., flip) the same set of channels to mislead the voting based fusion technique. We denote I_{attack}, as the fraction of channels that a selfish node flips/changes from its observed vector. Unlike the attacks from malicious nodes, selfish nodes attack on fewer but some specific channels (based on certain statistical or other criteria).

7.3 Gathering Channel Centric Evidence

For evidence collection, we use the same technique as we discussed in Section 4.2.1, where bounds on the predicted power levels for each neighbor is calculated on any channel k using the anomaly monitoring technique described by Fig. 7.1. But the evidence collection in the prior section was node centric, where we had an evidence vector for each neighbor. In contrast, now we collect the same evidence vector but for each channel k, the total number of channels being K.

Consider Fig. 7.1. Let O be the position of any node i. Let j be its neighbor whose exact location is not known, but the mutual distance can be estimated through RSS localization. Through RSS localization, whenever node j sends an advertisement, we can estimate the mutual distance s_{ij} and the hence the locus of neighbor j is anywhere on the circle centered around node i with a radius s_{ij} when transmit powers are same for all nodes. (See Fig. 4.1) Using commonly used propagation model for RSS [43], we know RSS at node i on channel k due to primary tower T_k is:

$$\gamma_k^i = P_k \times \frac{A^2}{s_{i,k}^{\omega}}; \tag{7.1}$$

where A is a constant, ω is path loss factor, $s_{i,k}$ is the distance between T_k and node i, and P_k is the transmit power of T_k. On any channel k, the highest and lowest bounds on the received power for neighbor j due to the primary transmitter T_k transmitting on channel k is given by: $[\gamma_k^j]_{high}$ and $[\gamma_k^j]_{low}$. It has been shown in Section 4.2.1, that the upper and lower bounds are:

$$[\gamma_k^j]_{high} = P_k \times \frac{A^2}{s_{j_{min},k}^{\omega}} \tag{7.2}$$

Thus the predicted power vector for node j on channel k as calculated by i is given as:

$P_{predict}^j = \big[\ ([\gamma_1^j]_{low}, [\gamma_1^j]_{high}), \cdots, ([\gamma_k^j]_{low}, [\gamma_k^j]_{high}), \cdots, ([\gamma_K^j]_{low}, [\gamma_K^j]_{high})\big]$.

The inference drawn by node i for node j on channel k is given as

$$d_k^j|_{infer} = \begin{cases} 0 & \text{if both } [\gamma_k^j]_{low} \text{ and } [\gamma_k^j]_{high} \leq \gamma_{th} \\ 1 & \text{if both } [\gamma_k^j]_{low} \text{ and } [\gamma_k^j]_{high} \geq \gamma_{th} \\ X & \text{otherwise} \end{cases} \tag{7.3}$$

where X denotes that no inference could be drawn, and γ_{th} is the common threshold used by all nodes to decide if a channel is occupied or not.

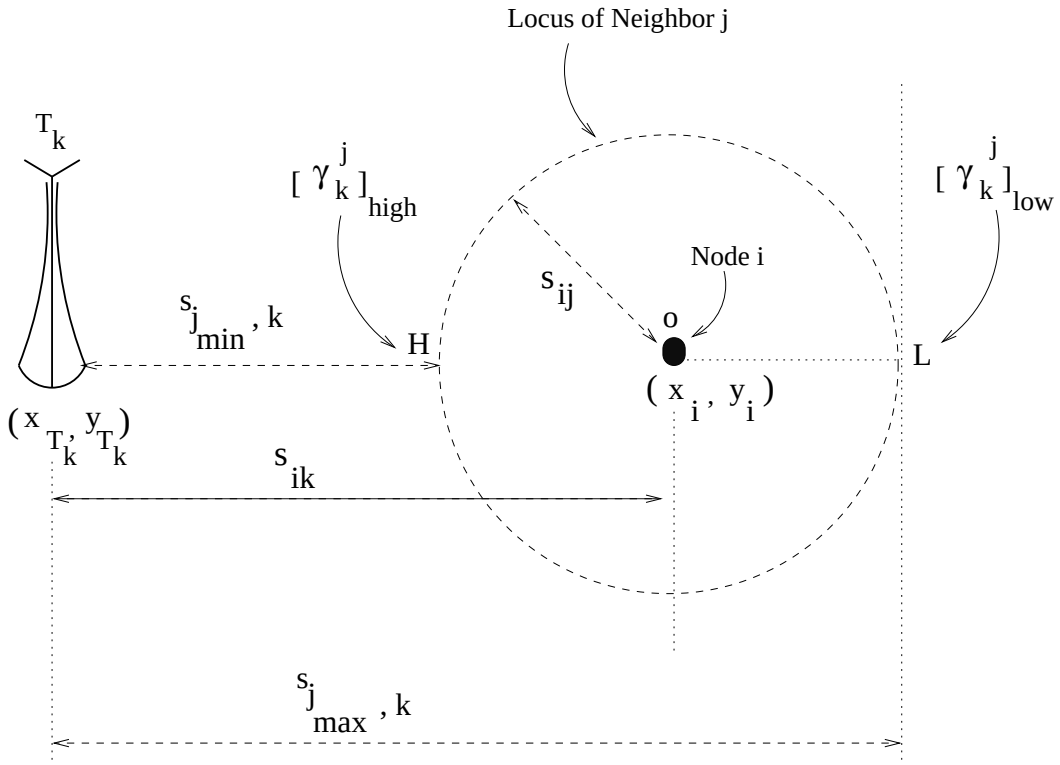


Figure 7.1: Bounds of RSS on channel k of neighbor node j

7.3.1 Formation of trust evidence

The predicted occupancy vector, given the mutual distance between node i and j, is given as

$$D_i^j = [d_1^j|_{infer}, ..., d_K^j|_{infer}]; \quad d_k^j|_{infer} \in \{0, 1, X\} \tag{7.4}$$

We compare D_i^j from Eqn. (7.4) with received $B_{adv}^j = [d_1^j, .., d_k^j..d_K^j]$. At any node i, we record the results of comparison in a vector *Trust Evidence* Q_k^j, where a match (denoted as α), a mismatch (denoted by β), and channels with value X in D_i^j are recorded as μ based

on Eqn. (7.5). If Q_k^j is the result of the comparison, then

$$Q_k^j = \begin{cases} \alpha & \text{if } d_k^j|_{infer} = d_k^j \\ \beta & \text{if } d_k^j|_{infer} \neq d_k^j \\ \mu & \text{otherwise} \end{cases} \tag{7.5}$$

From Eqn. (7.5), for each channel k and for each neighbor j, we have one of the three categories of outcomes: match, mismatch, undecided. We arrange the Q_k^j for all neighbors and look for total number of occurrences for each category with respect to channel k. The number of matches, mismatches and undecided observed for channel k are given by n_{α_k}, n_{β_k}, and n_{μ_k} respectively. Also, $n_{\alpha_k} + n_{\beta_k} + n_{\mu_k} = N_k$ equals the total number of opinions from N neighbors on channel k. Note this is equally valid for all nodes and therefore we drop the index i for the node concerned.

7.4 Bayesian Framework for Trust based Channel Preference

From the previous section, we get N independently monitored observations on channel k, comprising one of the three possible outcomes per observation. We seek to obtain the Bayesian belief (also called subjective probability) of occurrence of each possible outcome known as *Bayesian belief parameters* based on prior observations gathered from the observed trust evidence. With more observations, we update the Bayesian belief parameter for the hy-

pothesis increasing the accuracy of the parameters. Since a match indicates a non-anomalous behavior, a channel with higher posterior belief for match is considered more trustworthy.

To model how node i can compute the belief on channel k, we use the observation counts to calculate the Bayesian estimate of each of the parameters. Since the following analysis is valid for any channel k, we drop the suffix k for simplicity. Let $X(\bar{\theta})$ denote the hypothesis described by the underlying unknown Bayesian probability parameter of a random trial yielding match, mismatch or undecided as $\bar{\theta} = \{\theta_\alpha, \theta_\beta, \theta_\mu\}$. Here, θ_α, θ_β, and θ_μ are the unknown probability of $X(\bar{\theta})$ exhibiting a match, mismatch or undecided respectively. Since these observation outcomes are exhaustive and mutually exclusive, $\theta_\alpha + \theta_\beta + \theta_\mu = 1$. Let D_α, D_β, D_μ denote the random variables that represent the number of times, the outcomes α, β and μ occur. The observation data can be represented as random observation vector $D(N) = \{D_\alpha, D_\beta, D_\mu\}$ having a multinomial distribution with 3 tuple parameter described by θ_α, θ_β, and θ_μ.

Our objective is to estimate and update the probability parameters in $X(\bar{\theta})$ based on observation evidence $D(N)$ and prior information on the hypothesis parameter $\bar{\theta}$, itself. In pursuit, we these we propose the following theorem in Section 7.4.1.

7.4.1 Bayesian Inference based Decision Reliability

Since there is no information about $\bar{\theta}$ initially, we consider it to be uniformly distributed a-priori. Subsequent observations dictate how these parameters are updated. Our first step is to calculate the Bayesian estimate of $\bar{\theta}$.

First, we show the case of estimating belief that a match occurs (θ_α). Since in Bayesian inference, the assumption is that prior and posterior probability have the same distribution, we can formally define the probability parameters as:

$$\begin{aligned} P(X(\bar{\theta}) = \alpha|\bar{\theta}) = \theta_\alpha \\ P(X(\bar{\theta}) = \beta|\bar{\theta}) = \theta_\beta \\ P(X(\bar{\theta})) = \mu|\bar{\theta}) = \theta_\mu \end{aligned} \tag{7.6}$$

This assumption is due to the fact that a Dirichlet distribution acts as a conjugate prior to multinomial distributions. Hence prior and posterior preserve the same form [50].

The observations data $D(N)$ can be treated as a multinomial distribution with probability parameter θ_α,θ_β, and $\theta\mu$, where the probability mass function is given by:

$$P(D_\alpha = n_\alpha, D_\beta = n_\beta, D_\mu = n_\mu|\bar{\theta}) = P(D(N)|\bar{\theta}) = \frac{N!}{n_\alpha! n_\beta! n_\mu!} \theta_\alpha^{n_\alpha} \theta_\beta^{n_\beta} \theta_\mu^{n_\mu}$$

Given this we can use Bayes theorem to calculate the posterior belief estimate on the event of a match $\hat{X}(\bar{\theta}) = \alpha$, given observation data $D(N)$ as:

$$P(\hat{X}(\bar{\theta}) = \alpha | D(N)) = \frac{P(X(\bar{\theta}) = \alpha, D(N))}{P(D(N))} \tag{7.7}$$

Denominator of the above equation is the marginal probability that can be conditioned or marginalized on all possible outcomes for $\bar{\theta}$ and since probabilities are continuous

$$P(D(N)) = \int_{D(N)(\bar{\theta})} P(D(N)|\bar{\theta}) f(\bar{\theta}) d(\bar{\theta}) \tag{7.8}$$

Since there is no prior information on $\bar{\theta}$ (before any observations) in Eqn. (7.8), we can assume it to be uniformly distributed such that $f(\bar{\theta}) = 1$ and we can put Eqn. (7.7) in Eqn. (7.8), and get

$$P(D(N)) = \frac{N!}{n_\alpha! n_\beta! n_\mu!} \int_{D(N)(\theta_\alpha, \theta_\beta, \theta_\mu)} \theta_\alpha^{n_\alpha} \theta_\beta^{n_\beta} \theta_\mu^{n_\mu} d\theta_\alpha d\theta_\beta d\theta_\mu \tag{7.9}$$

For simplicity, let $\int_{D(N)(\theta_\alpha, \theta_\beta, \theta_\mu)} \theta_\alpha^{n_\alpha} \theta_\beta^{n_\beta} \theta_\mu^{n_\mu} d\theta_\alpha d\theta_\beta d\theta_\mu = I_1$

To solve for I_1 we use the multivariate generalization of the Eulerian integral of first kind. Note that $D(N)(\theta_\alpha, \theta_\beta, \theta_\mu)$ denotes a space and we know that a space of $m(= 3)$ parameters has only $m - 1(= 2)$ degrees of freedom due to the additivity constraint $\theta_\alpha + \theta_\beta + \theta_\mu = 1$. Therefore when we integrate over this space, the integration has $m - 1 = 2$ dimensions.

Hence

$$I_1 = \int_0^1 \int_0^{1-\theta_\alpha-\theta_\beta} \theta_\alpha^{(n_\alpha+1)-1} \theta_\beta^{(n_\beta+1)-1} (1-\theta_\alpha-\theta_\beta)^{(n_\mu+1)-1} d\theta_\alpha d\theta_\beta \tag{7.10}$$

Eqn. (7.10) is a known form for the multivariate extension of the Beta function which in this case is defined as $B(n_\alpha+1, n_\beta+1, n_\mu+1)$. The proof can be found in Lemma 2.4.1 of [51]. In general $B(\alpha_1, \cdots, \alpha_m)$

$$\begin{aligned} &= \int_{D(x_1,\cdots,x_{m-1})} x_1^{\alpha_1-1} ... (1-\sum_{i=1}^{m-1} x_i)^{\alpha_m-1} dx_1 ... dx_{m-1} \\ &= \frac{\prod_{i=1}^m \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^m \alpha_i)} = \frac{\Gamma(\alpha_1)\cdots\Gamma(\alpha_m)}{\Gamma(\alpha_1+\cdots+\alpha_m)} \end{aligned} \tag{7.11}$$

Using the above result, we can write Eqn. (7.10) as

$$\begin{aligned} I_1 &= B(n_\alpha+1, n_\beta+1, n_\mu+1) \\ &= \frac{\Gamma(n_\alpha+1)\Gamma(n_\beta+1)\Gamma(n_\beta+1)}{\Gamma(n_\alpha+1+n_\beta+1+n_\mu+1)} \end{aligned} \tag{7.12}$$

Putting Eqn. (7.12) in Eqn. (7.9) we get:

$$P(D(N)) = \frac{N!}{n_\alpha! n_\beta! n_\mu!} \frac{\Gamma(n_\alpha+1)\Gamma(n_\beta+1)\Gamma(n_\beta+1)}{\Gamma(n_\alpha+1+n_\beta+1+n_\mu+1)} \tag{7.13}$$

Since the parameters in gamma functions $n_\alpha + 1$ etc. are all non zero positive values, we can use the result $\Gamma(z) = (z-1)!$ to calculate Eqn. (7.13) as

$$P(D(N)) = \frac{N!}{(N+2)!} \tag{7.14}$$

Assuming conditional independence between the $\hat{X}(\bar{\theta})$, $D(N)$ and $\bar{\theta}$, we calculate the numerator of Eqn. (7.7), $P(\hat{X}(\bar{\theta}) = \alpha, D(N))$, as:

$$\begin{aligned}
&= \int_{D(N)(\bar{\theta})} P(X(\bar{\theta}) = \alpha, D(N)|\bar{\theta}) f(\bar{\theta}).d(\bar{\theta}) \\
&= \int_{D(N)(\bar{\theta})} P(X(\bar{\theta}) = \alpha|\bar{\theta}) P(D(N)|\bar{\theta}) d(\bar{\theta}) \\
&= \frac{N!}{n_\alpha! n_\beta! n_\mu!} \int_{D(N)(\theta_\alpha,\theta_\beta,\theta_\mu)} \theta_\alpha \theta_\alpha^{n_\alpha} \theta_\beta^{n_\beta} \theta_\mu^{n_\mu} d\theta_\alpha d\theta_\beta d\theta_\mu \\
&= \frac{N!}{n_\alpha! n_\beta! n_\mu!} \int_{D(N)(\theta_\alpha,\theta_\beta,\theta_\mu)} \theta_\alpha^{n_\alpha+1} \theta_\beta^{n_\beta} \theta_\mu^{n_\mu} d\theta_\alpha d\theta_\beta d\theta_\mu
\end{aligned} \tag{7.15}$$

The above integral has the same form as Eqns. (7.9), (7.10), and (7.11). Hence the integral portion of Eqn. (7.15) can be rewritten as

$$= \int_0^1 \int_0^{1-\theta_\alpha-\theta_\beta} \theta_\alpha^{(n_\alpha+2)-1} \theta_\beta^{(n_\beta+1)-1} (1-\theta_\alpha-\theta_\beta)^{(n_\mu+1)-1} d\theta_\alpha d\theta_\beta$$

which can be solved using Eqn. (7.11).

Using the above result, Eqn. (7.15) can be simplified as

$$P(\hat{X}(\bar{\theta}) = \alpha, D(N)) = \frac{N!(n_\alpha + 1)}{(N+3)!} \tag{7.16}$$

Thus, Eqn. (7.7), can be solved by dividing Eqn. (7.16) by Eqn. (7.14), which gives

$$P(\hat{X}(\bar{\theta}) = \alpha | D(N)) = \frac{n_\alpha + 1}{N+3} \tag{7.17}$$

Similarly, $P(\hat{X}(\bar{\theta}) = \beta | D(N)) = \frac{n_\beta+1}{N+3}$ and $P(\hat{X}(\bar{\theta}) = \mu | D(N)) = \frac{n_\mu+1}{N+3}$. These equations are the expressions for posterior belief of matches, mismatches, and undecided. For any channel k, we again use the suffix k, for example n_{α_k}. For simplicity of notations, we rewrite the left hand side of Eqn. (7.17) for channel k as $R_{\alpha_k} = \frac{n_{\alpha_k}+1}{N_k+3}$; $R_{\beta_k} = \frac{n_{\beta_k}+1}{N_k+3}$; and $R_{\mu_k} = \frac{n_{\mu_k}+1}{N_k+3}$ respectively. Of course, it can be verified that $R_{\alpha_k} + R_{\beta_k} + R_{\mu_k} = 1$ and it satisfies the Cromwell's rule.

7.4.2 Trust updates

A node observes channels over several time slots. Let at any time t the number of matches, mismatches, and undecided for channel k are $n_{\alpha_k}(t)$, $n_{\beta_k}(t)$ and $n_{\mu_k}(t)$ respectively. The total number of neighbors advertising on channel k at time t is $N_k(t)$. The time window after which a trust update is made may vary depending on the user requirement and computational resource availability. The system under question may have selfish nodes that employ

static attacks, where channels attacked remain the same over time, or a dynamic variation, where channel sets that are attacked may change with time. If we assume a dynamic variation of collaborative attack, then channel attacked in previous slot may not be attacked in subsequent slots, and since channel usage is based on the current spectrum scenario, the weightage to old observations should be minimal. Hence we propose an update model with high sensitivity to latest observations and low sensitivity to old observations. This not only addresses dynamic attacks but automatically consider static attacks.

We also recommend that the length of the window for a trust update should be small, so as to capture frequent changes in the attacked channel sets. However, it may also be noted that frequent changes in collaborative channel sets increase the cost of attack for the adversary. Hence it is unlikely that the channel sets attacked will be changed on every slot.

Let the current time slot be denoted as t_n. The updated belief corresponding to match on a channel k at time t_n is

$$R_{\alpha_k}^{t_n} = \frac{1 + \sum_{t=1}^{n} \lambda^{t_n - t}\ n_{\alpha_k}(t)}{3 + \sum_{t=1}^{n} \lambda^{t_n - t} N_k(t)} \tag{7.18}$$

where $0 < \lambda < 1$ is a sensitivity factor that determines the extent to which old observations are weighted. $t = 1$ denotes the oldest observation and $t = n$ denotes the latest. Smaller values of λ implies that old observations have less weight and vice-versa. $R_{\beta_k}^{t_n}$ and $R_{\mu_k}^{t_n}$ can be calculated in similar ways from Eq. (7.18).

7.4.3 Net trust

To capture the effect of both the updated beliefs for matches as well as mismatches into a single parameter, we define *net trust* as the difference between $R_{\alpha_k}^{t_n}$ and $R_{\beta_k}^{t_n}$. More formally,

$$R_{net_k}^{t_n} = R_{\alpha_k}^{t_n} - R_{\beta_k}^{t_n} \tag{7.19}$$

Note, the range of $R_{net_k}^{t_n}$ lies between -1 and 1. We use net trust to rank channels from the best to worst. The greater the posterior belief for mismatch, the lesser is its net trust. This is consistent with the idea that mismatches indicate dishonest opinions. Hence we expect channels that were attacked to have significantly lower trust.

7.5 A Generalized Bayesian Multinomial Framework for Cooperative Decision Reliability

Until, now we have been focusing on proposing trust models from evidence gathered for a specific network. However, we seek to show that our proposed multivariate Bayesian trust model can be extended to any cooperative network decision making system where some feedback or anomaly monitoring system is in place. In that light, we provide the following discussion that shows how we can seamlessly integrate our earlier model discussed in Section 7.4, into any generic network system where the problem is to quantify availability or reliability of a decision making process.

Reliability of a cooperative decision mechanism is critical for the proper and accurate functioning of a networked decision system. Usually the individual nodes of a network send their inputs (votes) or opinions to a centralized or distributed decision maker who fuses inputs from different nodes according to a fusion rule. However, adversaries may choose to compromise the inputs from different sets of nodes that comprise the system. Often times, the monitoring mechanisms fail to accurately detect compromised inputs (due to inherent imperfection or environmental variations); hence cannot categorize all inputs into polarized decisions: compromised or not compromised. Oftentimes, reliability of a cooperative decision in a networked system depends on how well the individual nodes perform and how reliable they are [52]. To improve reliability of a decision making process in the presence of possible component failures, redundancy and voting schemes are often used [53]. Usually voting schemes are associated with a fusion rule (e.g., majority or plurality voting) which dictates how individual votes are combined to decide the final output.

A malfunction or a security breach due to an adversary might result in a faulty input (vote) to the central decision making entity. Such inputs may potentially have an adverse effect on the reliability of the decision process– the extent of which depends on the inputs in question. A failure monitoring mechanism is supposed to detect such faults and provide a '*feedback*' on each component. However, this monitoring mechanism might not have the ability to identify such faults with certainty due to inherent imperfect temporal and spatial factors. Hence a binary decision on whether a fault occurred or not is not always possible for all inputs. With the adversary employing a wide variety of attacks to compromise a *dynamic*

set of inputs (i.e., deciding the number of inputs and which ones), and imperfect monitoring conditions, it becomes difficult for a failure monitoring mechanism to quantify reliability of a collective decision based on inputs from individual nodes.

In the presence of adversaries the compromised inputs may induce incorrect results in a cooperative decision making process. The set of inputs attacked usually vary over time. The damage is evident when some central entity of the system using some fusion rule fuses the inputs from all nodes– compromised or not. A simple example of a cooperative decision process could be a voting system where nodes vote to produce a binary outcome. If majority of the inputs from the nodes are compromised, then the simple majority voting rule may produce a wrong result [54].

Given this, we investigate the quantification of reliability of a cooperative decision process based on the inputs from various nodes. We assume, each component provides a single input, all of which are prone to attacks. The underlying imperfect failure monitoring mechanism produces varying feedback over time. We consider that the outcome of the monitoring mechanism can be placed into three categories: those we know have *not been compromised*, those we know have *been compromised*, and those which cannot be inferred either way. Given these, we compute the reliability of a decision process in making a decision i.e., how reliable its output is. In this regard, we conceptualize the outcome of monitoring the input over time as a multinomial hypothesis of a Bayesian inference model with three parameters. We build a Bayesian inference based reliability model, where we assign a value to a decision that indicates how reliable the outcome of the decision is. The way reliability

is computed also depends on how much risk a system can tolerate. For example, a mission critical system may need to have a more strict reliability model because the associated risks are too high. Therefore, we propose two ways of computing the reliability– first is an optimistic one and the second is a conservative one. The optimistic model could be applied to systems where some tolerance for wrong decisions are allowed. However, for a mission critical system where there is almost no room for erroneous decisions, the conservative model could be used. To account for the confidence associated with the reliability computation, we propose an entropy based uncertainty value to represent how certain or uncertain the computed reliability is. A lower uncertainty associated with reliability value is an indication of being more confident about the end decision.

7.5.1 Cooperative decision system model

We consider a time-slotted system comprising N voting nodes each of which provides only one input (i.e., the vote) on each time slot. The nature of the decision is generic; it could be as simple as a binary voting or it could be some complex decision metric. A centralized controller fuses all votes from each component through a fusion scheme (e.g., majority or plurality voting rule) to arrive at a *global decision*.

Adversarial model: We assume that all the inputs from each component are exposed to an *adversary* whose goal is to disrupt the voting process at the central controller. The adversary has some predefined attack resources and can choose to attack different sets of

inputs over time and also attack varying number of inputs in each time slot. However, it maintains a long term average of the fraction of the inputs it attacks which we call the probability of attack and denote as P_a. For example, $P_a = 0.6$ means that the adversary compromises 60% of the inputs over a large period of time. Hence a single observation (over one time slot) is not sufficient for characterizing the behavior of the adversary.

Imperfect component failure monitoring: We assume that there is a component failure monitoring or anomaly detection mechanism in place that infers whether the input from each component has been compromised or not. Oftentimes, the monitoring mechanism cannot infer anomaly with certainty. Thus, it classifies the inputs into three categories: i) compromised, ii) not compromised, and iii) undecided. All three are a function of environmental parameters that may be dynamic over time. Note, the monitoring occurs over a period of time. Also system transients and noisy environments may increase temporal uncertainty. Therefore, reliability is computed over time– a larger time window of observation allows a more accurate estimation of the actual reliability.

Uniformly distributed prior inference: Since there is no bias over any of the three possible outcomes of the monitoring process, we assume that the initial probabilities of each is equal.

Probability of detection: We define the probability of detection as the percentage of 'nodes' inputs that can be *accurately* inferred as compromised or not compromised and denote it as P_{det}. Let us further illustrate the meaning of P_{det} using Fig. 7.2 that shows an

input in reality could be either compromised or not compromised. If compromised, it can be inferred as either as compromised with a probability (say a_1) or undecided with probability $1 - a_1$. Note, we assume that there is no way a compromised input will be inferred as not compromised. This assumption is because we argue that 'undecided' is inferred in absence of credible evidence else there will be a compromised feedback only when sure. Similarly, if an input is not compromised, it can be inferred as either 'not compromised' with a probability (say b_1) or 'undecided' with probability $1 - b_1$. Again, there is no way a not compromised component will be inferred as compromised. Thus, for the two real cases, detection occurs with probabilities a_1 and b_1. If an input has equal chances of being compromised and not compromised, then $P_{det} = \frac{a_1+b_1}{2}$. Else, a_1 and b_1 will have to be weighted with their corresponding probabilities. For all practical purposes, we consider P_{det} to be at least 0.5.

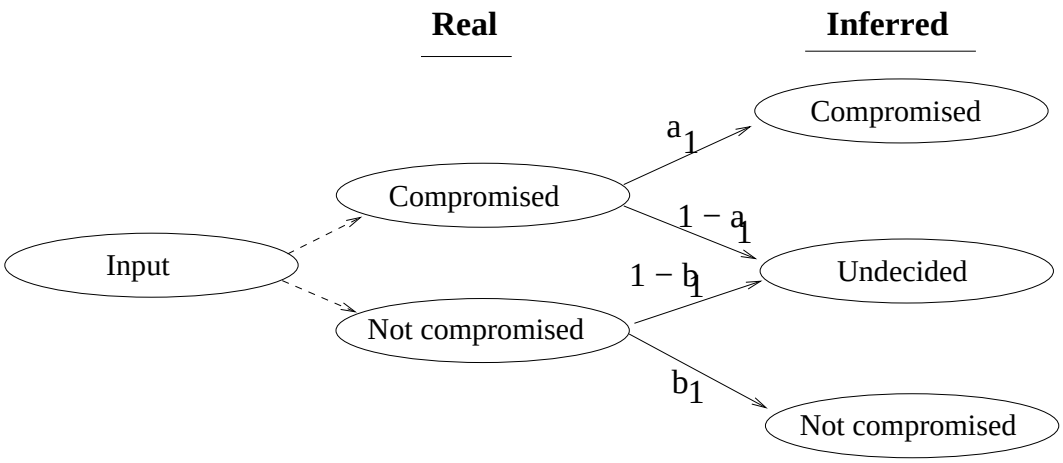


Figure 7.2: Inference possibilities for detection probability

The above features make the problem of computing the reliability of the output decision a probabilistic concept. Hence, we compute the reliability as a continuous process based on observations over time slots. If the adversary uses the same attack strategy, then

the reliability will converge sooner. On the other hand, if the adversary changes its attack strategy (i.e., dynamic attack strategy), the reliability will oscillate even for large time windows.

7.5.2 Reliability based decision making

Suppose that the three feedbacks of the monitoring mechanism– 'not compromised', 'compromised', or 'undecided' be denoted by α, β and μ respectively. Let n_α represent the number of inputs that have 'not' been compromised, n_β represent the number of inputs observed to have been compromised, and n_μ is the number of inputs for which we do not know either way. Of course, $n_\alpha + n_\beta + n_\mu = N$. Since the values of n_α, n_β and n_μ change over time, we represent these observations at time t as $n_\alpha(t)$, $n_\beta(t)$ and $n_\mu(t)$.

Since the system's underlying parameters of cooperative voting behavior are unknown, we propose to use Bayesian inference to update corresponding probability estimate for a hypothesis that the decision process is correct with a certain reliability. The system is only as reliable as the individual inputs are. Therefore, we have to calculate the posterior probabilities associated with encountering each of the three feedbacks. The final decision reliability will be a function of these posterior probabilities which are also known as belief estimate in Bayesian inference.

To begin with, an uniform belief over the three possibilities is assumed as there is no initial information. As time progresses, we update the belief estimate based on the observed

values of α, β, and μ which increases the accuracy of the estimate of the belief associated with each category.

We define θ_α, θ_β, and θ_μ as the probabilities for an input being 'not compromised, 'compromised', and 'undecided' respectively. Of course, $\theta_\alpha + \theta_\beta + \theta_\mu = 1$, since the outcomes are exhaustive and mutually exclusive. We define $X(\bar{\theta})$ as the hypothesis described by these underlying unknown *Bayesian probability parameters* where $\bar{\theta} = \{\theta_\alpha, \theta_\beta, \theta_\mu\}$.

Let D_α, D_β, and D_μ represent the random variables that represent the number of times the outcomes α, β and μ occur. The observation data can be represented as random observation vector $D(N) = \{D_\alpha, D_\beta, D_\mu\}$ having a multinomial distribution also known as *concentration hyperparameter* of the underlying 3-tuple probability parameter described by θ_α, θ_β, and θ_μ.

As mentioned earlier, there are N independently monitored nodes of a system whose parameters for voting behavior are unknown due to changing adversarial attack strategies and the imperfect monitoring mechanism. Given this, we calculate the Bayesian belief associated with 'not compromised'. Similarly, we will model Bayesian posterior belief for the other two cases as well viz. compromised and undecided.

We use the observation counts from the sequential observations over time to calculate the posterior Bayesian estimate of each of the parameters. Our objective is to estimate and update the probability parameters in $X(\bar{\theta})$, viz. θ_α, θ_β, and θ_μ based on observation evidence $D(N)$ and prior information on the hypothesis parameter, $\bar{\theta}$, itself.

Since there is no information about $\bar{\theta}$ initially, we consider the prior parameters of $\bar{\theta}$ to be uniformly distributed a-priori. Subsequent observations will decide how these parameters are updated. Our first step is to calculate the Bayesian estimate of $\bar{\theta}$.

First, we show the case of estimating belief that a 'not compromised' occurs (θ_α). We can use our prior framework by using Eqn. 7.17. In a similar way, we can calculate θ_β and θ_μ. These equations are the expressions for posterior belief of 'not compromised', 'compromised', and 'undecided'. To simplify the notations of belief estimates of the three categories, we rewrite them as R_α, R_β, R_μ respectively. Of course, it can be verified that $R_\alpha + R_\beta + R_\mu = 1$. The above equations also satisfy the Cromwell's rule [55], which suggests that no prior belief unless logically impossible, should be assigned zero probability even if no events in that category has occurred so far.

We mentioned that in Bayesian inference posterior probabilities may be used as priors for future calculations. The Cromwell's rule leaves open the probability however small, to experience an event that has not occurred yet but may happen in future. Hence Bayesian estimates should have non-zero priors for an event that has not occurred yet. From the derived equation it is evident that even if $n_\beta = 0$, $R_\beta \neq 0$, is a very small number but not zero when N is large.

7.5.3 Reliability models

We propose two reliability models and also show how entropy can capture the uncertainty associated with the reliability calculation due to the undecided inputs.

7.5.3.1 Decision Reliability for an Optimistic System

For a system, we assumed that adversary has uniformly chosen the inputs it chooses to attack i.e., there is no reason for preferential attack on a certain component's input. Hence we can account for the undecided R_μ by splitting it in the ratio of $R_\alpha : R_\beta$, and adding it to the value R_α to provide the optimistic reliability denoted by R_s^o. Of course, when the proportion of R_μ is high, we may not be as confident on the reliability value than when we have lower values of R_μ. Thus, R_s^o is computed as:

$$R_s^o = R_\alpha + \frac{R_\alpha}{R_\alpha + R_\beta} R_\mu \tag{7.20}$$

7.5.3.2 Decision Reliability for a Conservative System

Unlike the optimistic approach, where the undecided ones are split in a ratio, the conservation model treats the undecided ones as if they were compromised. In other words, only the 'not compromised' event (R_α) is used for computing the reliability. Hence,

$$R_s^c = R_\alpha \tag{7.21}$$

This conservative way of computing the reliability is more appropriate for mission-critical systems where the decisions can only be made based on the 'not compromised' inputs. No risk is taken on the undecided inputs even if there could be some that were not compromised.

7.5.3.3 Uncertainty associated with Reliability

System reliability as computed by Eqn. (7.20) can yield the same value for different sets of R_α, R_β, and R_μ. For example, consider two scenarios.

Scenario 1: $R_\alpha = R_\beta = 0.5$, and $R_\mu = 0$

Scenario 2: $R_\alpha = R_\beta = 0.3$, and $R_\mu = 0.4$

For both scenarios, the optimistic decision reliability as given by Eqn. (7.20) is 0.5 as shown in Table 7.1. It can be noted that though scenario 2 has higher R_μ than scenario 1, R_s^o for both are the same. However, intuitively we ought to trust scenario 1 more than scenario 2 because more certain decisions have been made when R_μ is less. We know that higher values of R_μ reduces the chances of being closer to the real value of reliability. As shown in the two scenarios, the optimistic reliability can yield the same value for different sets of R_α, R_β, and R_μ

In order to illustrate the uncertainty associated with the R_μ, we use entropy which is a measure of uncertainty inherent in a system. Usually, entropy of a system uses the steady state probabilities that the system could be in. We define the entropy of the decision reliability as:

$$E_s = -[R_\alpha log_2(R_\alpha) + R_\beta log_2(R_\beta) + R_\mu log_2(R_\mu)] \tag{7.22}$$

The entropy, E_s, captures the uncertainty which is shown in Table. 7.1. Scenario 1 with $R_\mu = 0$ has an uncertainty measure of 0.69 which is lower than scenario 2 with 1.08. Thus, the reliability of scenario 1 can be trusted more than that of scenario 2.

Now, let us consider scenario 3 which has the same R_μ as of scenario 2; however, R_β is higher than scenario 2. Hence scenario 3 has lesser reliability value than scenario 2. Since scenario 2 has been established to be less trustworthy than 1, we can also verify that from R_s^o of scenario 3 is also less than that of scenario 1. However, these arguments hold true only when at least 50% of the observations are accurate observations, i.e., $R_\mu < 0.5$. This is a reasonable assumption as it is impractical to have a detection mechanism which cannot detect or decide majority of the time.

The certainty measure can be represented by a value $C_s = 1 - E_s$, through which we can sort systems in terms of how confident we are about the corresponding reliabilities. The concept is useful to resolve a *tie* while choosing more reliable systems. Hence we conclude that scenario 1 is most reliable choice followed by scenario 2 and then scenario 3. System reliability

is hence a combination of optimistic reliability value and entropy measures associated with the evidence.

Table 7.1: Reliability-Entropy tuple; N=1000

Scenario	R_α	R_β	R_μ	R_s^o	E_s	C_s
1	0.5	0.5	0.0	0.5	1.00	0.00
2	0.3	0.3	0.4	0.5	1.57	−0.57
3	0.2	0.4	0.4	0.33	1.51	−0.51
4	0.7	0.3	0.0	0.7	0.88	0.12
5	0.10	0.6	0.3	0.14	1.29	−0.29

7.6 Modeling Decision Reliability under Errors: A Special Case

Although in the previous section we argued that a monitoring system decides an *outcome* to be undecided, if there is lack of sufficient evidence to point towards a polarized binary decision. However, some environments may be severely noisy (like wireless systems) where there may be errors. Hence there is a possibility that actually 'compromised' (α) may be erroneously decided as not compromised (β) and vice versa. The Fig. 7.3, shows such additional possibilities not described by model in Fig. 7.2. We define the modified probability of detection as the percentage of inputs that can be *accurately* inferred as compromised or not compromised and denote it as P_{detect}. Let us further illustrate the meaning of P_{detect} using Fig. 7.2 that shows an input in reality could be either compromised or not compromised. If compromised, it can be inferred as either as 'compromised' with a probability a_1 (correct) or ''undecided with probability a_2 (uncertain) or 'not compromised' with a probability a_3 (missed detection). Similarly, if an input was not compromised, it can be inferred as ei-

ther 'not compromised' with a probability b_1 (correct) or 'undecided' with probability b_2 or 'compromised with a probability b_3 (false alarm). Thus, for the two real cases, detection occurs with probabilities a_1 and b_1. If an input has equal chances of being compromised and not compromised, then $P_{detect} = \frac{a_1+b_1}{2}$. Else, a_1 and b_1 will have to be weighted with their corresponding probabilities. For all practical purposes, we consider P_{detect} to be at least 0.5, since it is impractical to quantify reliability where majority of feedbacks are incorrect. Similarly, $P_{uncertain} = \frac{a_2+b_2}{2}$ denote the probabilities of ignorance (expressing inherent uncertainty) about inputs. $P_{error} = \frac{a_3+b_3}{2}$ denotes probability of errors made by the feedback system. These probabilities are used for performance evaluation.

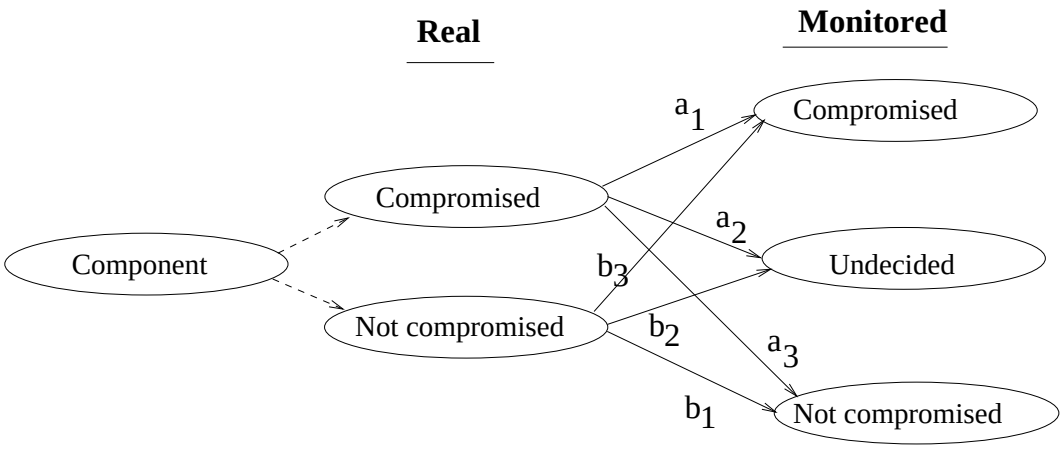


Figure 7.3: Special Case: Inference possibilities under errors

CHAPTER 8: SIMULATION MODEL AND RESULTS

In this chapter we represent an exhaustive collection of simulation results for performance analysis of our proposed optimistic and conservative trust models in terms of identifying malicious nodes, improving accuracy of cooperative sensing under attacks, analyzing performance of different strategies employed from malicious nodes perspective, and channel centric bayesian approaches towards quantifying reliability.

8.1 Simulation Set Up

In order to validate the proposed trust models, we conduct extensive simulation experiments. For optimistic trust model and training data sets, we simulate a CR network with 30 randomly distributed secondary nodes out of which 9 are programmed malicious. The nodes are scattered over 60x60 area with a sharing radius of 20 units. There are 40 primary transmitters each corresponding to one channel. The operating spectrum length for most results is 40 channels. For some results we considered denser networks with the same number of nodes in a smaller area. We choose the other simulation parameters from [56], where the normalization threshold is kept at -33.7dbm and transmit power of primary is 1000 mW. The pathloss factors are varied from 3 to 6. For our testing set, for identifying malicious nodes

we choose a static network of 100 nodes in a 600X600 grid and one third of them are chosen to be randomly malicious and the sharing radius is kept as 200 units. We subject our model to different attack strategies, pathloss and radio environments and different attack measures. For inversion based fusion, we choose a network of 100x100 with 100 nodes keeping other factors same. Most of the results are for non collaborative SSDF. We will compare our fusion results with the blind majority voting fusion rule, majority voting based exclusion and KL divergence based techniques.

8.2 Attack Strategies by Malicious Nodes

First, we compare the two attack measures P_{attack} and I_{attack} to investigate how they affect the cooperative spectrum sensing result in Section 8.2.1. We find the *percentage of mismatches* from the ideal fusion result which is used as a performance analysis measure. We report which out of the two is a better attack measure from malicious user's perspective; i.e., the greater the percentage of mismatches, the better would be the attack measure. The reverse would be true when we analyse results from defender's perspective.

Second, we seek to find that given an attack measure, how the collaborative or non-collaborative SSDF strategies affects the network using the same performance analysis measure in Section 8.2.2.

8.2.1 Intensity of attack versus probability of attack: Effects on cooperative sensing accuracy

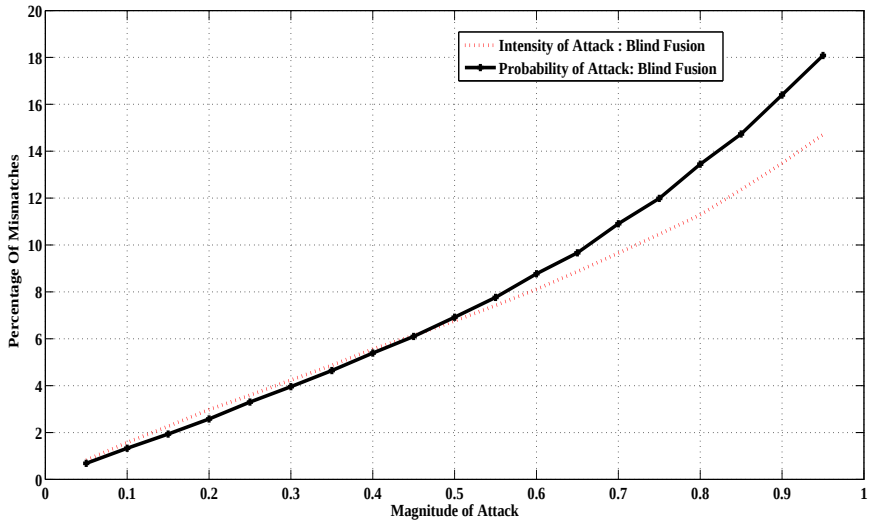


Figure 8.1: Percentage of mismatches with blind fusion using I_{attack} and P_{attack}

We plot the performance when nodes blindly fuse local spectrum sensing data from all nodes, (Blind Fusion), for different values (magnitude of attack) under P_{attack} and I_{attack} as shown in Fig. 8.1. The results throw light upon which out of the two is a better attack measure from the malicious user's perspective and the relationship of percentage of mismatches (the damage effect) with the magnitude of attack. We observe that P_{attack} is a better attack strategy when the magnitude of attack is greater than 0.5. When the values of the attack measures are less than 0.5, there is little difference on the damages they inflict in terms of deviation from the actual result. In addition, P_{attack} does not require any extra effort hence is feasible from the cost perspective. Hence, we conclude employing P_{attack}

is more useful for the malicious nodes (given majority voting is used for fusion), and hence most of the subsequent results assumes the worst case attack measure P_{attack} unless explicitly mentioned.

8.2.2 Non-collaborative versus collaborative SSDF: Effects

We plot and compare the performance analysis measure when nodes use blind majority voting fusion under collaborative and non collaborative attack strategies for different values of P_{attack}. The results elucidate which is a better strategy from the malicious user's perspective and their relationship with value of P_{attack}. From Fig. 8.2, we observe that for most values of P_{attack}, collaborative SSDF is a better attack strategy in terms of the deviations it inflicts from the actual result. However, collaborative SSDF becomes less effective than its non collaborative counterpart when $P_{attack} > 0.8$. Hence the conclusion is given $P_{attack} < 0.8$, collaborative SSDF is a better attack strategy. However, it must be noted that cost of collaboration between malicious nodes is higher than non collaboration and may not always be feasible. For most of the subsequent results, we assume non collaborative SSDF unless explicitly mentioned. We keep broader study under collaborative SSDF as future work.

In Fig. 8.2, we compare the damages done by collaborative SSDF attack and non collaborative SSDF attack. We observe that collaborative attack is able to damage more for most P_{attack}, except when $P_{attack} > 0.80$. Hence if cost, synchronization and fading issues are taken care of collaborative attacks will cause more damage for lower magnitudes of P_{attack}.

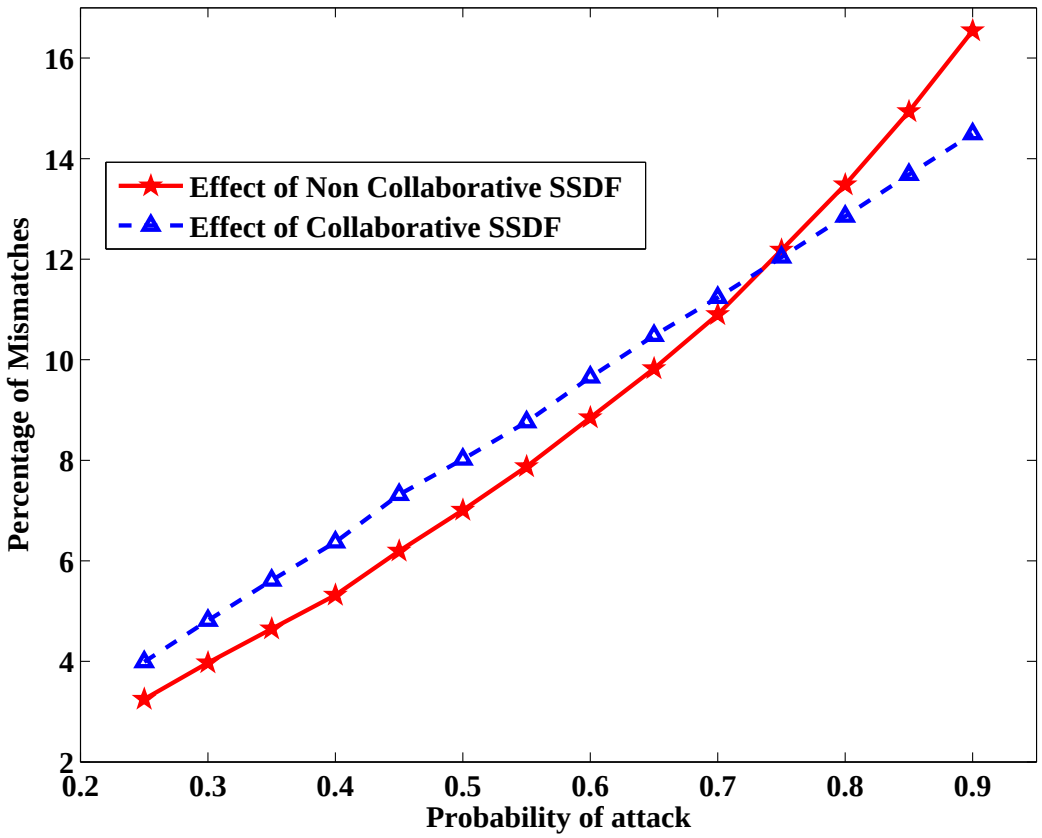


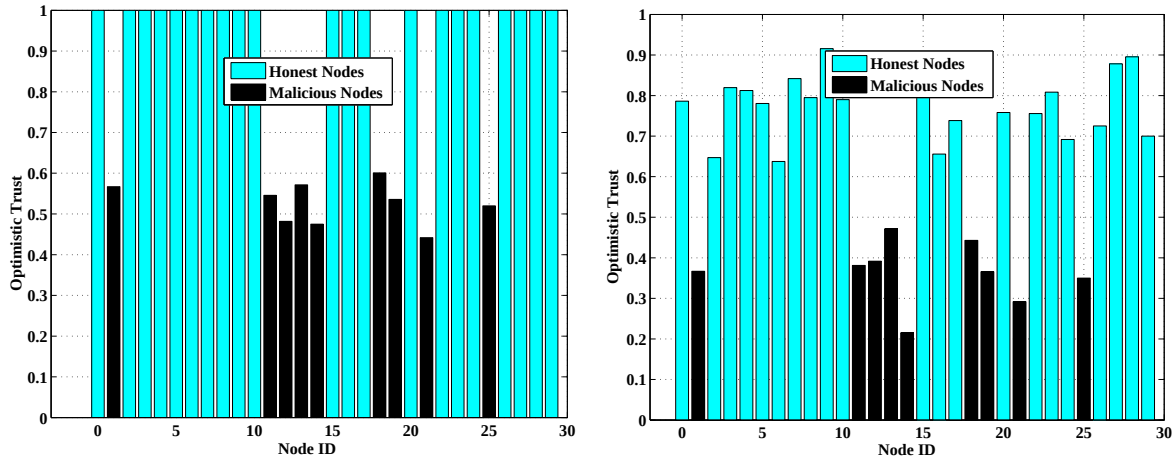
Figure 8.2: Comparison of effects: Collaborative vs non-collaborative SSDF

8.3 Optimistic Trust Model: Trust Measurement

In this section, we show the results of trust computation for the optimistic trust heuristic and the modified beta trust. We also show relevant differences between steady state and transient values.

8.3.1 Optimistic trust heuristic: Transient state

We measure the transient state trust values for the optimistic trust heuristic. Fig. 8.3(a) depicts the trust distribution when all malicious nodes have a $P_{attack} = 0.4$ and there is no control channel noise. We observe that the honest nodes have very high (= 1) trust value, and malicious nodes have much lower trust values. In Fig. 8.3(b), we capture the trust values under a very noisy environment, where bits in B_{adv} are randomly flipped or are lost (with $P_{attack} = 0.50$). From the figures, it is evident that there is a significant difference between the trust values of honest and malicious nodes although the honest nodes do not have trust values as high as those in Fig. 8.3(a).



(a) (b)

Figure 8.3: Optimistic Trust Heuristic (a) For $P_{attack} = 0.40$ (b) For $P_{attack} = 0.50$

8.3.2 Optimistic trust heuristic: Steady state

Next, we find the steady state and the instantaneous values for a malicious node under the two attack measures $P_{attack} = 0.50$ and $I_{attack} = 0.50$. Fig. 8.4 shows the instantaneous trust and moving average trust of a node over time as seen by one of its neighbor under I_{attack}. Fig. 8.5 shows the same but when the node uses P_{attack}. We see that in the former case, the convergence of the average trust value reaches quicker (on time slot 40) than in the case of when P_{attack} is launched where convergence is reached after 80 time slots. This is because, P_{attack} attacks different number of channels as well as on different channels keeping a long term mean while I_{attack} attacks on different channels. Hence I_{attack} in general can be detected or learned quicker. This is an incentive that may discourage malicious entities to use I_{attack}. The variance of instantaneous trust value from the mean is 5 times higher in P_{attack} than that of I_{attack}.

8.3.3 Intensity vs. probability of attack: The better alternative

From Figs. 8.4 and 8.5, we provide a qualitative explanation on which is a better attack measure if the malicious nodes want to stay undetected for the maximum amount of time. For I_{attack}, total number of channels remain same everytime although individual channels attacked vary. P_{attack} induces more doubts where the number of channels as well as channels attack vary. Hence the second strategy is far more difficult for defenders to form a concrete

opinion quickly. The second attack strategy will have more standard deviation and variance when it comes to the distribution of instantaneous trust samples.

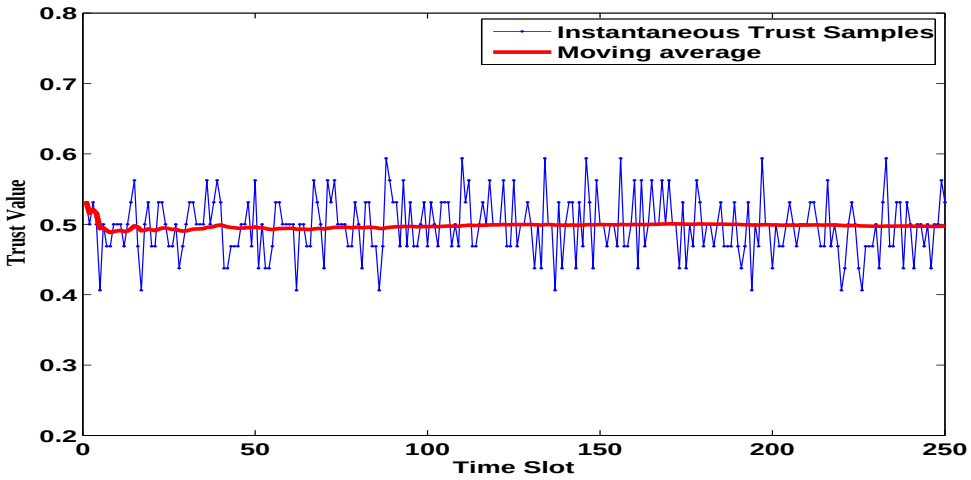


Figure 8.4: Optimistic trust heuristic over time under I_{attack}: Instantaneous and moving average for a node as seen by neighbor

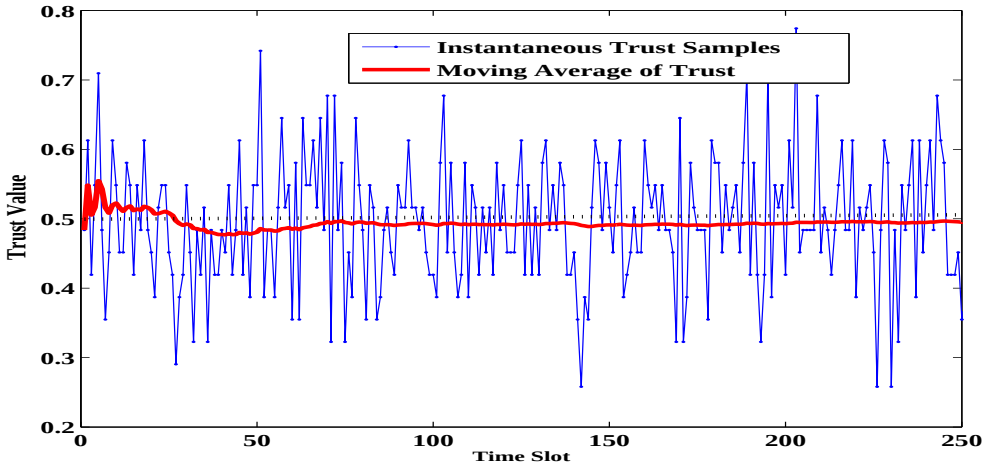
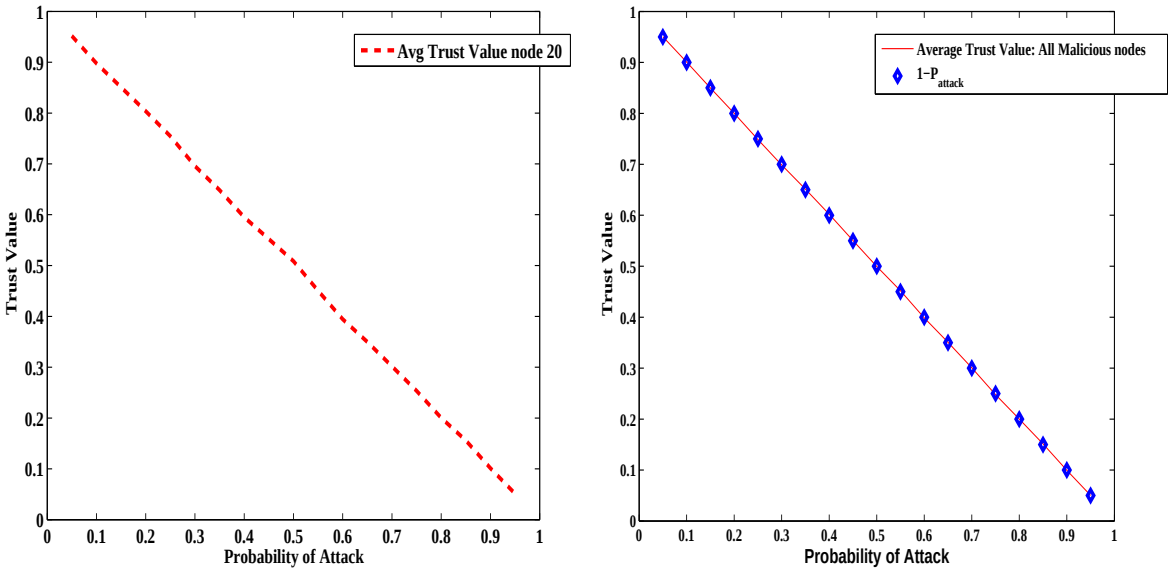


Figure 8.5: Optimistic trust heuristic over time under P_{attack}: Instantaneous and moving average

8.3.4 Relation of optimistic trust values and magnitude of attack

Given that there is no noise during transmission of advertised vectors and randomly chosen channels sets, we conclude that the relation between optimistic trust heuristic and magnitude of attack is an inverse linear relation. In general, $E^{j,i}_{steady} \approx 1 - P_{attack}$ or $E^{j,i}_{steady} \approx 1 - I_{attack}$. This is true for an individual malicious node as seen in Fig. 8.6(a), and for the set of all malicious nodes as seen in Fig. 8.6(b), over different magnitudes of attack. In general, we conclude that trust value decreases as the attack magnitude grows. Thus we prove that the more a malicious node attacks the more it damages it's reputation in the network. This observation is intuitive as trust is an indication of the level of cooperation a node does to the network. If it attacks or does not cooperate with a probability of P_{attack}, then that node cooperates with a probability of $1 - P_{attack}$. Hence we conclude that the moving average malicious node's trust value converge to a value approximately equal to $1 - P_{attack}$ for the optimistic trust model. The advantage of this is that from the trust value we can learn the magnitude of attack which may have many potential uses. We do not recommend forgetting factors (except for on-off attack model) as used in many existing trust models, as given our attack model, use of forgetting or discounting moving averages induce loss of information on the true level of aggressiveness of a malicious node.



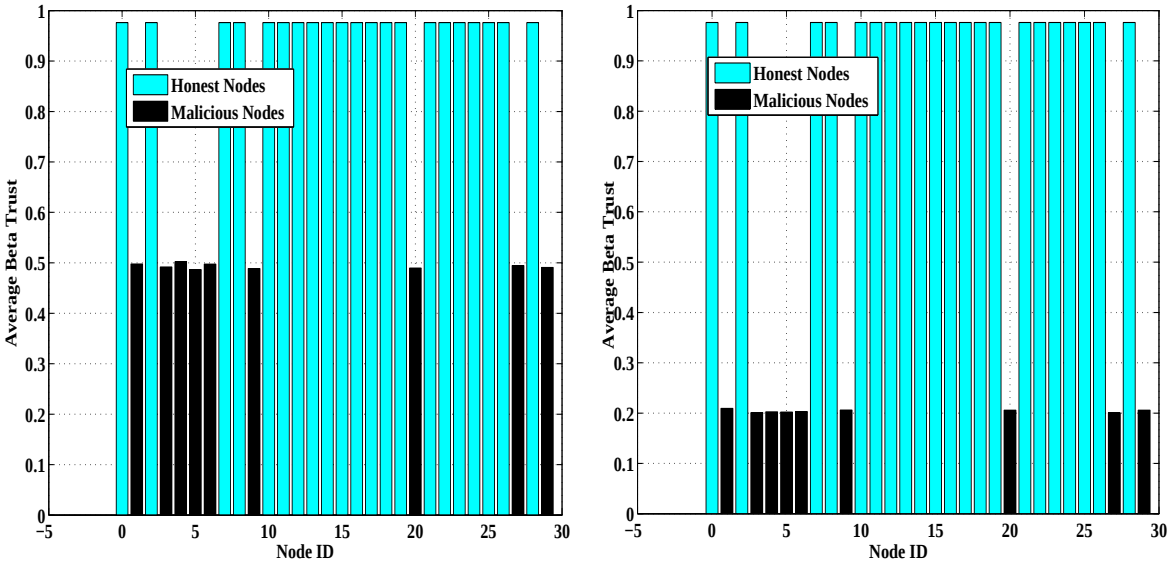
(a) (b)

Figure 8.6: Optimistic Trust over P_{attack} (a) Individual Node (b) Overall Network

8.3.5 Beta Trust as a subjective probability

In an effort to adhere to Cromwell's rule, and to depict the subjective probability interpretation of trust, we use Eqn 5.4 to compute the approximated Beta distribution based trust. The trust values are almost the same as the optimistic trust heuristic with the exception that honest nodes never have a value exactly equal to 1, but a very close to 1. Likewise, even if a node changed all the local sensing results, its value will be very close to 0, but not exactly equal to 0. Fig. 8.7(a), shows the approximated Beta trust based steady state values for honest and malicious nodes when $P_{attack} = 0.50$. Fig. 8.7(b), shows the beta expectation based trust values at steady state for honest and malicious nodes when $P_{attack} = 0.80$. The

relation between beta expectation based trust value and magnitude of attack is a sub-linear inverse relation in contrast to linear inverse for optimistic trust heuristic.



(a) (b)

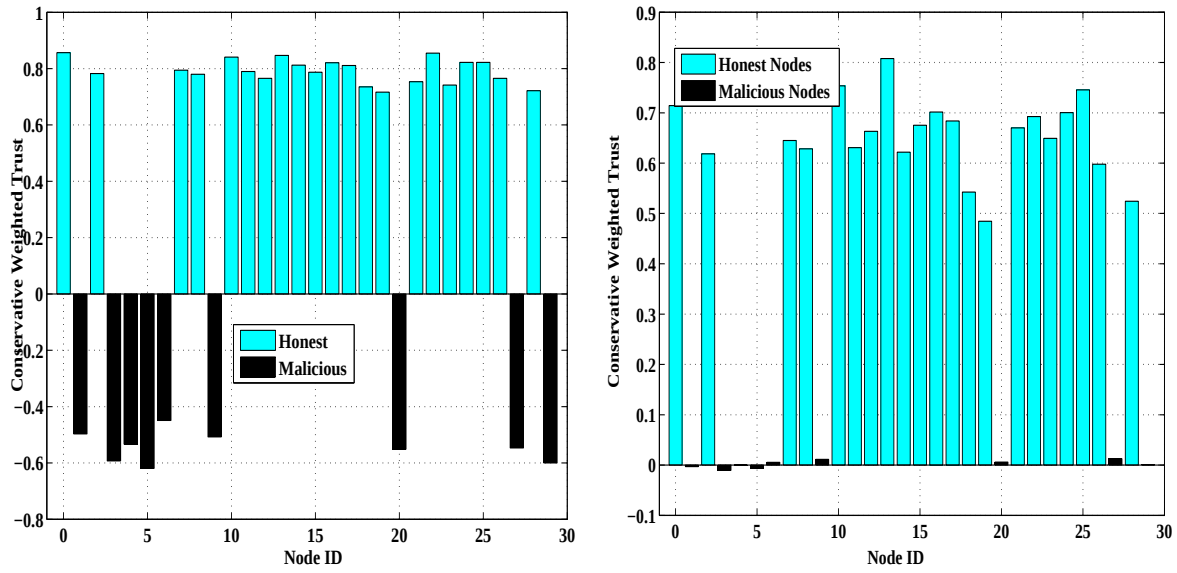
Figure 8.7: Environment with Pathloss=3:(a) Optimistic Trust for $P_{attack} = 0.50$ (b) Optimistic Trust for $P_{attack} = 0.80$

8.4 Conservative Trust Model: Malicious Node Identification

In this section, we discuss the results of conservative trust model. We look at the distribution of the conservative trust weights for different pathloss environments and different P_{attack} values. This will give idea as to how pathloss and P_{attack} effect the conservative trust model, and whether there is significant difference between honest and malicious nodes.

8.4.1 Computed trust values

Fig. 8.8(a), shows the conservative trust weights in an environment of pathloss factor 5 and $P_{attack} = 0.8$. We see that all the malicious nodes have high negative trust weights while honest nodes have positive trust weights. Hence the differentiation is evident. Using a network with 30 nodes with a sharing radius of 20 units scattered on an area 60x60 we show that the trust values of malicious nodes are correctly reflected even when local malicious node density (per honest node) is very high. This result is an improvement from voting based exclusion and joint entropy based exclusion as we show in Section 8.4.3.



(a) (b)

Figure 8.8: Conservative Trust Value: (a) Pathloss=5; $P_{attack} = 0.80$ (b) Pathloss=4; $P_{attack} = 0.50$

Figure. 8.8(b) shows the conservative trust weights in an environment of pathloss factor 4 and $P_{attack} = 0.5$. We see that all the malicious nodes either have very low positive or very low negative trust weights closer to zero, while honest nodes have positive trust weights. The difference between the trust values between malicious and honest nodes is evident, but not as drastic as the previous case when pathloss factor was 5 and $P_{attack} = 0.80$. The first reason for this trend is because an intermediate value of P_{attack} attacks fewer channels. Another difference from the preceding result, is that the weights of honest nodes are comparatively low in this case than the previous scenario. This is because pathloss 4 induces more uncertainty in evidence which is reflected in their lower trust than honest node values in Fig. 8.8(a).

Figure. 8.9 shows trusts for the preceding example with $P_{attack} = 0.5$ but a lower pathloss factor of 3. Through this, we seek to see the effect the pathloss factor alone, keeping P_{attack} same. We see that there is an increase in the weights of honest nodes while there is not much to choose for the malicious node values. However, if the $P_{attack} = 0.80$, the gap between the trust weights become evident as was depicted in Fig. 8.8(a). The reason for increased values of honest node's is due to less uncertainty induced due to lower pathloss factor 3.

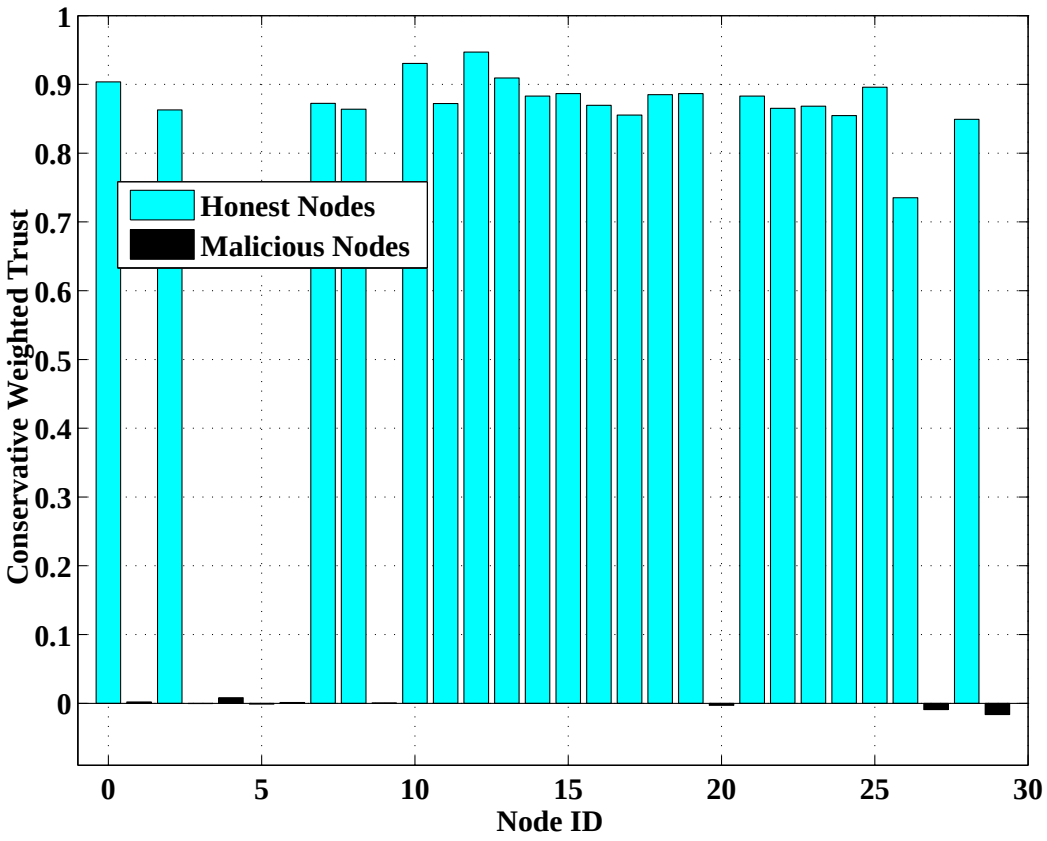


Figure 8.9: Conservative Trust Value: Pathloss=3; $P_{attack} = 0.50$

8.4.2 Worst case performance: An improvement from existing approaches

Fig. 8.10, we show that both the proposed models work well under high densities of malicious nodes (greater than 60%), and under collaborative SSDF attacks. This is in contrast to existing approaches that cannot detect when nodes are collaborative and should number of malicious nodes exceed 50% as we will show in Section. 8.4.3. We simulate a scenario with 11 honest nodes and 19 malicious nodes, ensuring the high average local densities of malicious node around each node. We calculate the conservative trust w_j for each node in

Fig. 8.10. From the figure, it is evident that our proposed method works even under a high malicious node density (63%) with collaborative attacks. Validation of accurate detection across different densities is shown in Fig. 8.11.

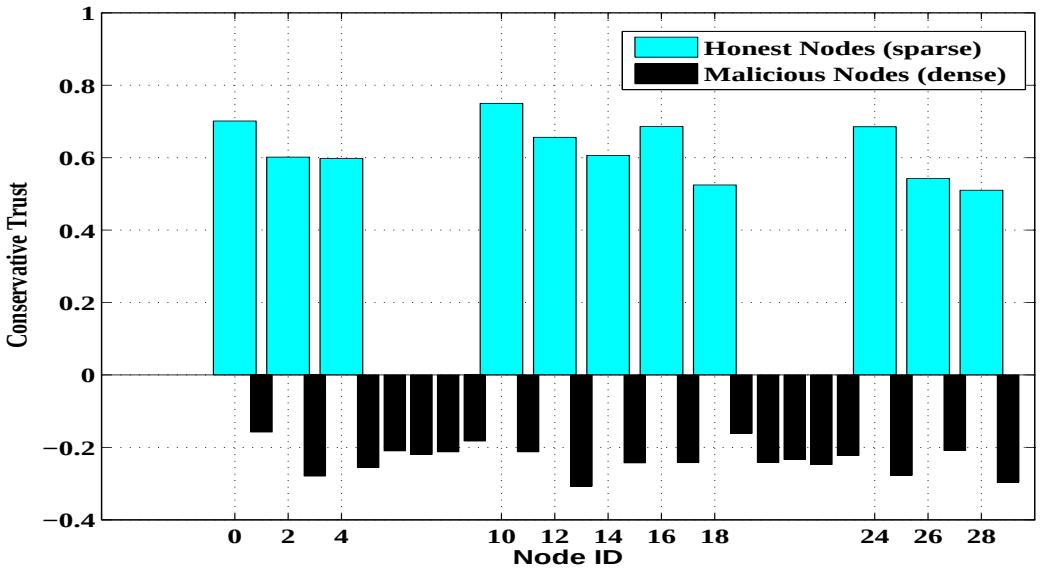


Figure 8.10: Worst case performance under high density of collaborative malicious nodes and high P_{attack}

8.4.3 Comparison of node identification with existing research

We seek to compare the benefits of our proposed trust based malicious node detection scheme with a few existing research works. We use the more comprehensive and sophisticated conservative trust model and compare with KL divergence and majority voting based exclusion (decouple) method. We compare the percentage of accurate detection over various fraction of malicious nodes for different P_{attack}.

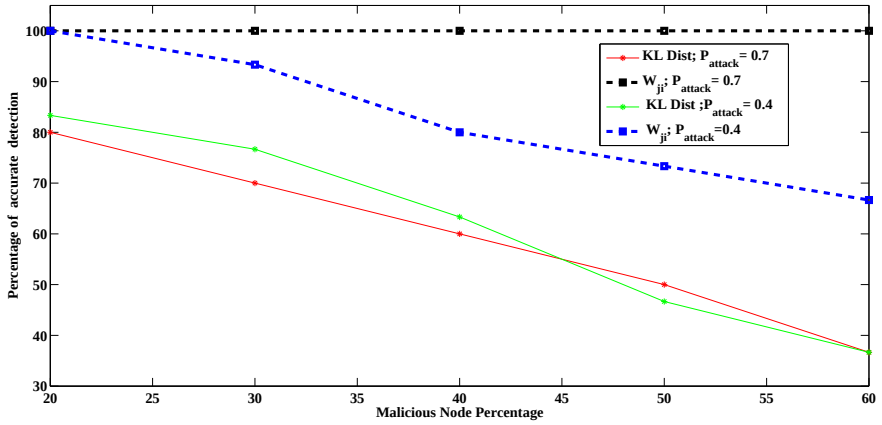


Figure 8.11: Comparison of Proposed Trust Models with KL distance method

In Fig. 8.11, we observe that under various values of fraction of malicious nodes and P_{attack}, our method using w_j value much better results than existing researches discussed in [36, 49] particularly under high fractions of malicious nodes or high P_{attack}.

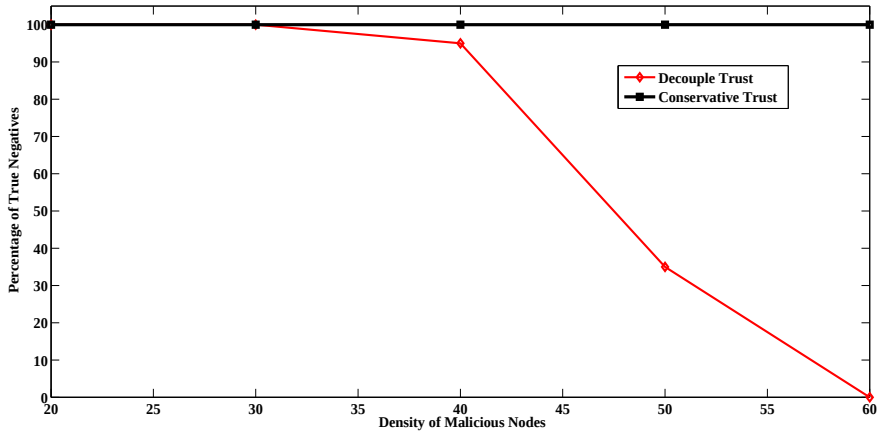


Figure 8.12: Comparison of Proposed Trust Models with majority voting based exclusion

In Fig. 8.12, we compare our work with another recent work [57]. We report significantly high true negative detection percentage across different malicious node fractions.

8.4.4 Testing set performance for malicious node identification

We consider a network with 100 nodes over an area of 600x600 where malicious nodes are initially chosen randomly that remain fixed throughout. The density of malicious nodes are kept at 0.25. The malicious nodes were equally divided into 3 groups where each group had different P_{attack}. The first group had a $P_{attack} = 0.30$, the second group had $P_{attack} = 0.50$, and the third group had a $P_{attack} = 0.80$. Given this, we apply our conservative threshold $w_{classify}$ to identify malicious nodes for a conservative system with different pathloss factors. We explore both options namely; different $w_{classify}$ based on the relevant environment and single $w_{classify}$ for all kinds of environments. By comparing, Figs. 8.13 and 8.14, we observe that the first approach gives more consistent results than the second approach for the group for malicious nodes with lower P_{attack} of 0.30. For $P_{attack} \geq 0.50$, either approach works well.

8.4.4.1 Testing Set Performance: Identification of malicious nodes with pathloss=4

From Fig. 8.13, we have the testing set for a CR network with pathloss factor 4 and the corresponding $w_{classify} = 0.29$. We observe that this effectively captures all 27 malicious nodes in our testing set. This is the result that mimics an ad-hoc CR network in a dense environment.

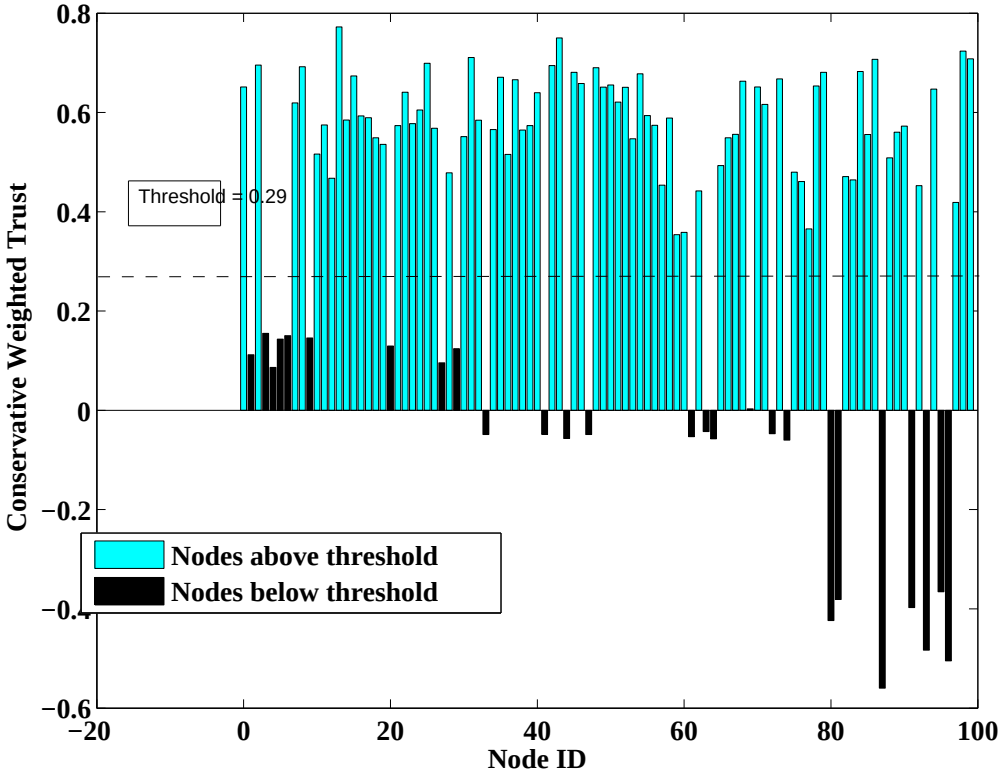


Figure 8.13: Testing set performance for node classification: Pathloss=4; $w_{classify} = 0.29$

8.4.4.2 Example bad performance for low P_{attack} under single $w_{classify}$

In Fig. 8.14, we seek to investigate whether we can apply $w_{classify} = 0.29$ to CR network with a different pathloss factor 3.2. We observe that this fails to capture 7 out 27 malicious nodes. These 7 nodes that are not detected are from the first group of 9 malicious nodes with low P_{attack} of 0.30. This is indicative that the approach of using a single $w_{classify}$ for classification is not optimal for minimizing missed detection. If we used a higher threshold, we would increase false alarms where we classify honest nodes as malicious. Hence we propose the use of different $w_{classify}$ specific to the environment.

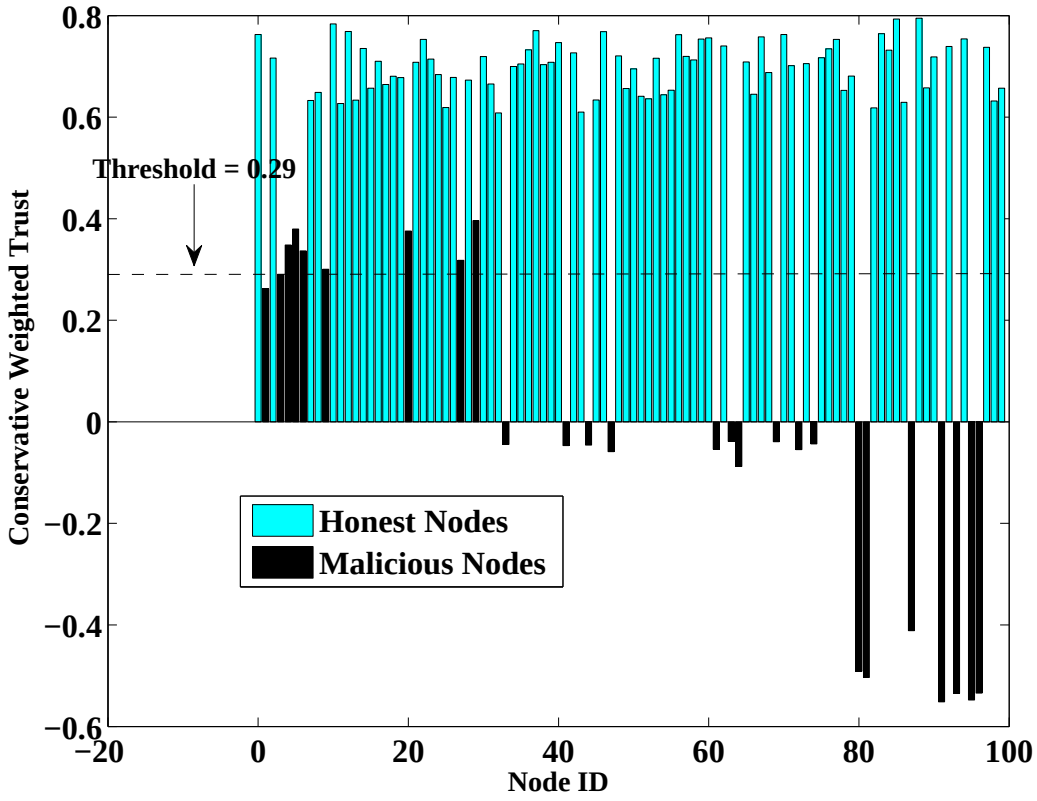


Figure 8.14: Missed detections using single $w_{classify}$: Pathloss factor = 3

8.4.4.3 Testing Set Performance: Identification of malicious nodes with pathloss=3.2

In Fig. 8.15, we have a testing set with pathloss factor 3.2. We use $w_{classify} = 0.42$ corresponding to the results of the SVM output corresponding to training set T_1 with pathloss 3 and $P_{attack} = 0.50$. We observe that this comfortably captures all of the 27 nodes as malicious. Hence the SVM results from the training set is able to generalize unseen examples for a different network.

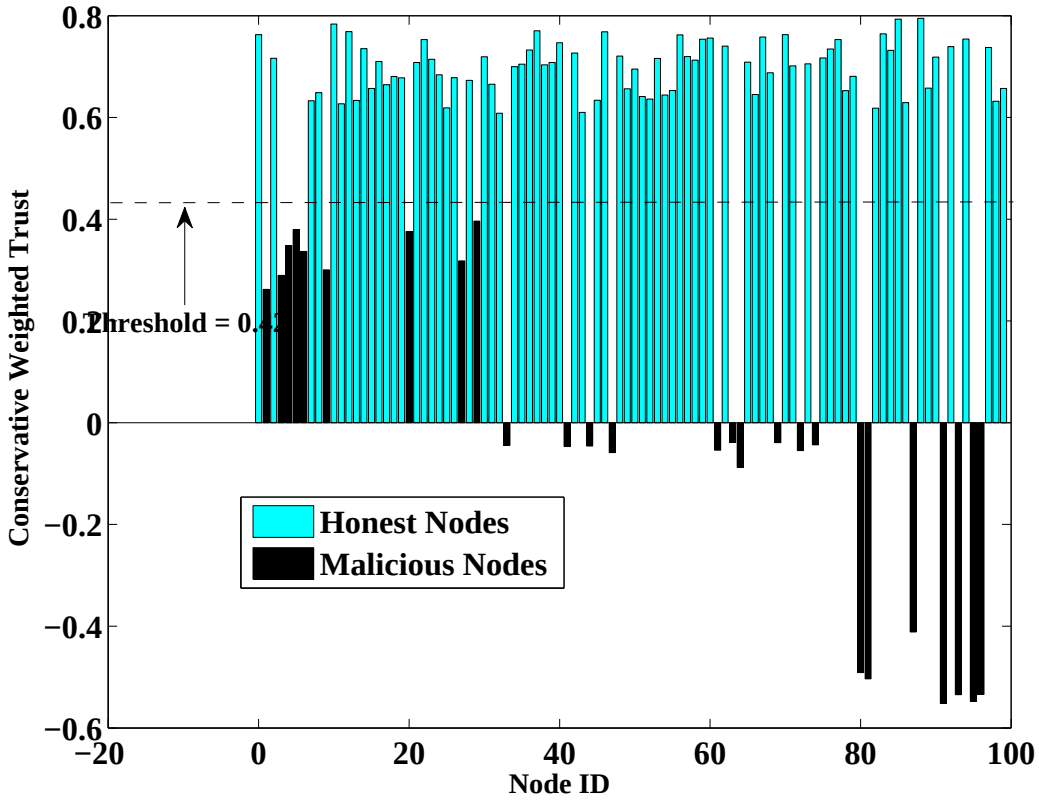


Figure 8.15: Testing set performance for node classification: Pathloss=3.2; $w_{classify} = 0.42$

8.4.4.4 Testing Set Performance: Identification of malicious nodes with pathloss=4.8

In Fig. 8.16, we have a testing set with pathloss factor 4.8. We use $w_{classify} = 0.39$ corresponding to the results of the SVM output corresponding to training set T_3 with pathloss 5 and $P_{attack} = 0.50$. We observe that this comfortably captures all of the 27 nodes as malicious. Hence the SVM results from the training set is able to generalize unseen examples for a different network in this case too.

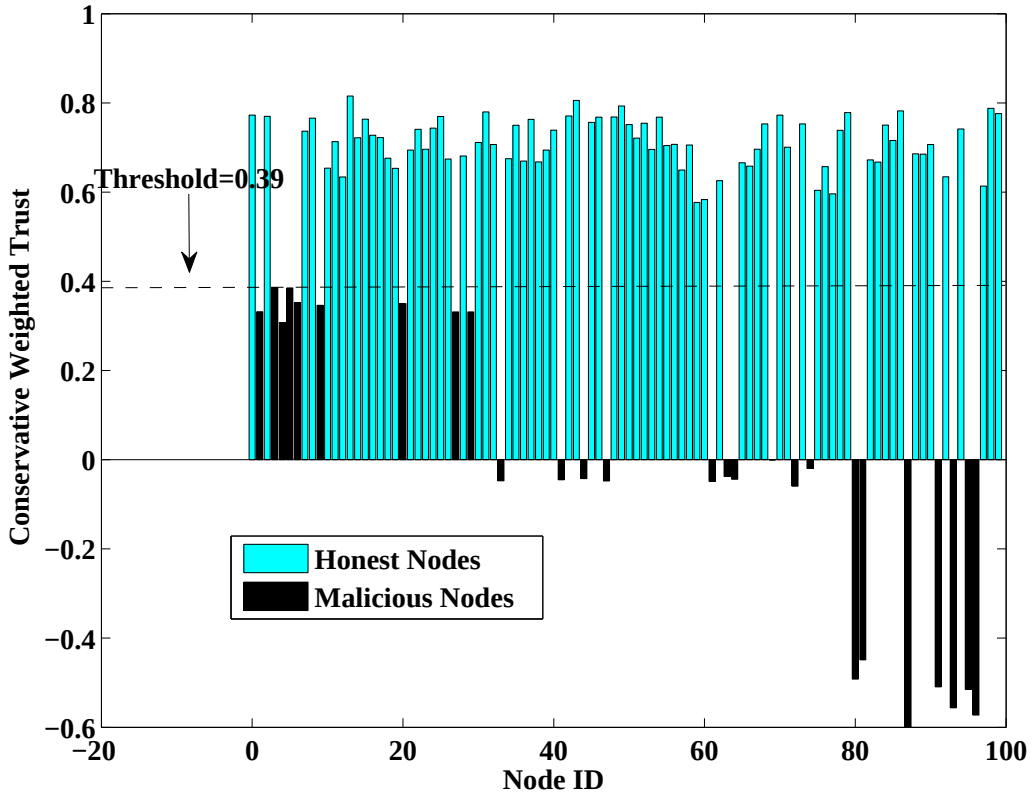


Figure 8.16: Testing set performance of node classification: Pathloss=4.8; $w_{classify} = 0.39$

In the above section, we report to have successfully identified malicious nodes under different pathloss environments and over different realistic values of P_{attack}.

8.4.5 Defending against on-off attacks: A special case

In On-Off attacks, we limit our study to particular node 20 which launches on-off attacks in the five stages over 500 slot time window as was discussed in Section. 6.2. We plot the results of On-Off attacks as seen by one of its neighbor node 29 using equations from the

asymmetric weighted moving average discussed in Section. 6.2. We compare the results with other popular trust update schemes and justify suitability of asymmetric averaging with regard to On-off attacks.

8.4.5.1 Choice of weighing factors and threshold

The weighing factors χ_a, $\chi_{b_{max}}$, $\chi_{c_{min}}$, and χ_d are chosen as 0.999, 0.999, 0.001 and 0.001. We can verify that this satisfies the conditions: $0 < \chi_{c_{min}} << \chi_{b_{max}} < 1$, $0 < \chi_a < 1$, and $0 < \chi_d < 1$. From the skewed values of the weighing factors $\chi_{c_{min}}$ and $\chi_{b_{max}}$, it justifies the asymmetry that we provide by giving negative behaviors a very high weightage and positive behavior and very low weightage after the first occurence of negative behavior. The choice of χ_a and χ_d can be used to control the rate of trust redemption. If a system requires slower trust redemption that lower value of χ_a and lower value of χ_d is necessary. We put these weighing factors in the four case based equations discussed in Chapter 6 in Section. 6.2. Since there is no particular magnitude of attack we keep the mid point between the trust value range $(-1, +1)$ as $\Gamma_{on-off} = 0$. However, Γ_{on-off} can be adjusted according to the requirements of the system. More conservative systems will have $\Gamma_{on-off} > 0$. Different values of χ_{min} and χ_{max} can be chosen to ensure more fairness to nodes in a network inherently susceptible to more bit flips due to noise.

8.4.5.2 Comparison with Equal Weighted Moving Average

In Fig. 8.17, we show how the proposed asymmetric weighted moving average performs as opposed to the equal weighted moving average. We observe that at Stage 1 with no attacks, both schemes preserve a high trust value, but when attacks start from the 101 st time slot for the next 50 slots, asymmetric weighted moving average ensures cumulative trust is decreased more rapidly and preserves a low value. Equal weighted moving average is slow to react due to the node having behaved well in the first 100 slots. This happens because once current value in a slot is less than zero, the model forgets previous high reputation through a very low value $1-\chi_{b_{max}} = 0.001$ and expresses extremely high weight $\chi_{b_{max}} = 0.999$ to the current values from the 101th time slot, thus causing the cumulative trust at stage 2 to decrease rapidly. In the beginning of Stage 3, when the attack ceases, we see that trust value reflected by asymmetric average is low enough (-0.25) to reflect node's malicious history while equal weighted moving average fails to capture because the on-off attack ratio is $1:2$, i.e., more slots with no attacks. This happens because, previous cumulative trust of less of than zero at the end of Stage 2 is given a very high weight compared to current honest behavior. It prevents the trust values to improve even during honest behavior. In Stage 4, when attack starts after honest behavior for 100 slots, we see the significant difference between the trust values of the two schemes is preserved. Same is the case in Stage 5, where the reasons of a very slow increase in trust values under asymmetric average compared to equal weighted average is the difference in weighing factors assigned to previous and current trust values. Hence, we conclude that asymmetric weights can offer the benefits not provided by equal

weighted moving average in terms of reacting quickly to on-off attacks and preserving a low trust value of a malicious node. Through this scheme we have ensured that even though it targets only 100 out of 500 slots, the model can identify such nodes.

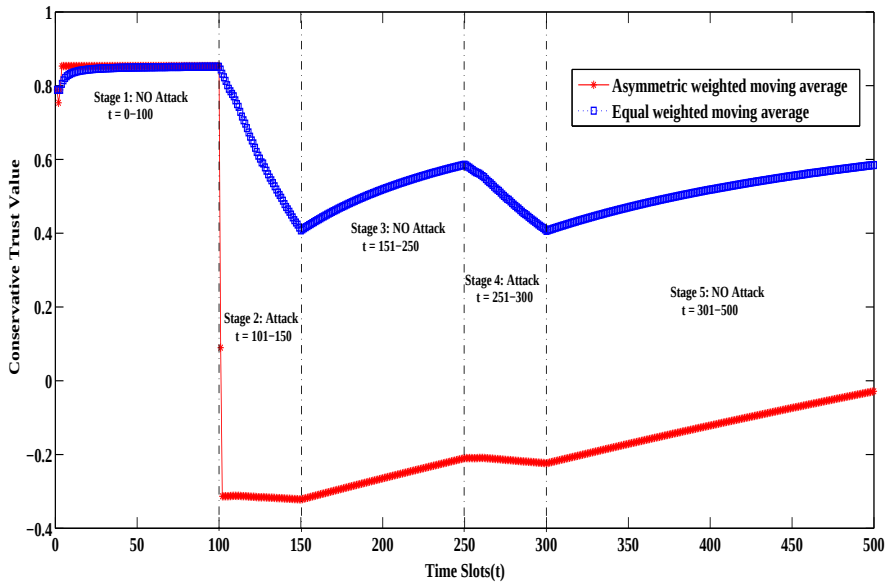


Figure 8.17: Asymmetric moving average vs equal weighted moving average: Node 20

8.4.5.3 Comparison with Exponential Weighted Moving Average

The major criticism of exponentially weighted moving average was that although it reacts quickly when attacks start, it also forgets malicious behavior as quickly as it reacts. This is inappropriate because a malicious node should not be allowed to increase its trust value quickly unless it engages in a long period of honest behavior to redeem its trust. The key point where a difference is created is case(c) of the on-off defense schema where we provide

very low value to honest behavior after a period of dishonest behavior. Hence it's cumulative trust value hardly increases. In Fig. 8.18, we do not see much difference in Stage 1 due to no attacks. Also there is not much difference is Stage 2 as there more weight given to new trust values by both models. However, in Stage 3, exponential weighted moving average allows the malicious node to quickly recover its trust value owing to forgetting old values. On the other hand, asymmetric average selectively does not forget old values that are low. This happens because, previous cumulative trust of less of than zero (selected Γ_{on-off}) at the end of Stage 2 is given a very high weight compared to current honest behavior. It prevents the trust values to improve even in the period of honest behavior. We see that for all subsequent stages the exponentially weighted averages oscillates between high and low values, but asymmetric average preserves a low value all the while at the same time allowing fairness by allowing very slow increase of cumulative trust at stage 5 owing to its continuous good behavior for 200 slots. This provision also helps nodes which experience noise to eventually redeem their trust on experiencing good transmission channels as we see next.

8.4.5.4 Distinguishing between On-Off attacks and Random Noise

The asymmetric model also provides some fairness associated for nodes which experience high temporal noise. This helps to differentiate malicious behavior from wireless channel induced effects as the node with temporal noise will eventually regain trustworthiness because noise is inherently bursty than on-off attacks. In Fig. 8.19, we can see results of a node modeled

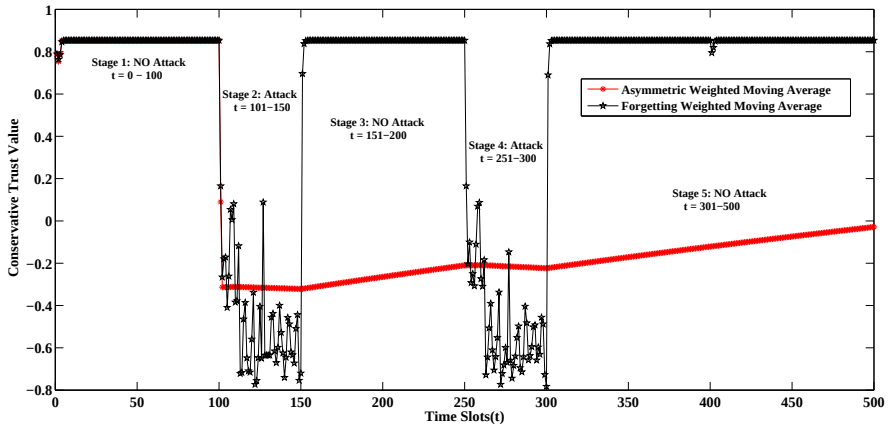


Figure 8.18: Asymmetric moving average vs. exponentially weighted moving average: Node 20

as an on-off attacker versus being an honest node experiencing noise. We see our approach that uses asymmetric average leaves an opportunity to redeem trust values decreased due to high temporal noise.

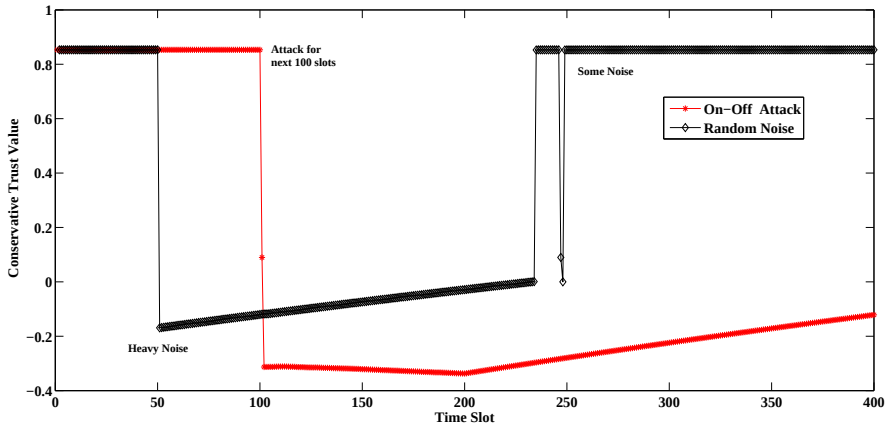


Figure 8.19: Difference of trust distribution due to malicious behavior vs random noise

8.4.5.5 Testing Set: Asymmetric average for nodes without on-off attacks

In practical systems, we may not know which node will launch on-off attacks and which nodes will not. So in this section, we study whether the asymmetric averaging is robust and generic enough to accurately preserve malicious identification for nodes which launch a regular P_{attack} instead of on-off attacks. In Fig. 8.20, we show the difference between equal weighted moving average and asymmetric moving average for the same node which launches $P_{attack} = 0.50$ instead of on-off attack in a testing set. We see that the correctness of asymmetric average update is not restricted to only nodes that launch on-off attack but also regular malicious nodes. Infact, asymmetric average is more conservative than other methods as it gives a much lower trust value that equal weighted moving average when a node launches probabilistic attack with 0.5 magnitude as seen in the figure below.

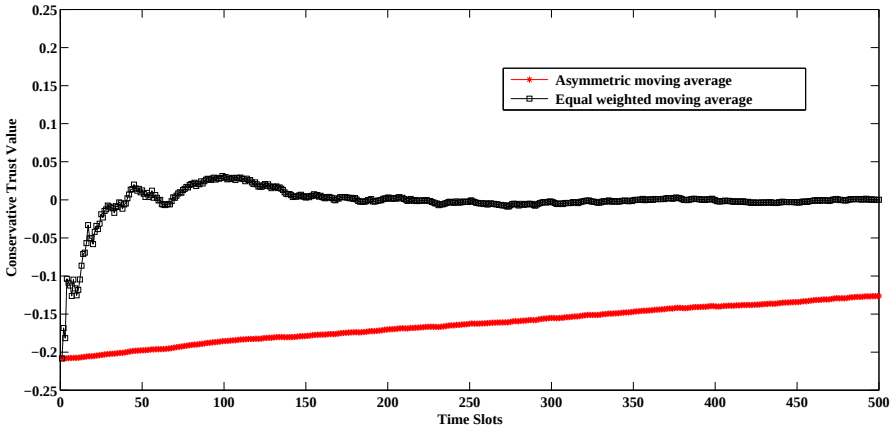


Figure 8.20: Asymmetric average for nodes without On-Off attacks:$P_{attack} = 0.5$

8.5 Trust based Fusion for Robust Decisions: Optimistic Trust Model

In this section we discuss all relevant result for trust based fusion using optimistic trust heuristic.

8.5.1 Optimal threshold

From Fig. 8.21, we obtain trust threshold Γ_{opt} for trust based fusion. We perform a trust based fusion with different candidate thresholds Γ ranging from 0.2 to 0.8 and compare how 'mismatch' varies from ideal results. Fig. 8.21, shows that for very low values of potential thresholds, there are more mismatches since most of the malicious nodes are included for fusion. However, as we increase this threshold, malicious nodes start getting discarded and mismatch decreases. However, when the threshold is very high (above 0.6), the mismatch again increases as high threshold means we are also discarding the potentially honest nodes too. Since our goal is to minimize the total average mismatch from the ideal scenario, we notice a range of threshold values from 0.45 to 0.51, where the average mismatch is the least for different probabilities of attack. From individual node's perspective, the idea is to minimize the mismatches after fusion. Hence we choose Γ_{opt} as 0.50, with the rationale that any node having trust value equal or below 0.50 is doing atleast more harm than it is doing good. If Γ_{opt} was 0.45, the nodes are successful in excluding majority of malicious nodes from fusion for higher probabilities of attack but not for lower probabilities of attack

because malicious nodes have lower trust values when they attack less aggressively. If we take 0.50 as threshold, then more malicious nodes are excluded. However, if we take $\Gamma_{opt} = 0.45$, we might end up being less strict by allowing more malicious nodes reports in fusion.

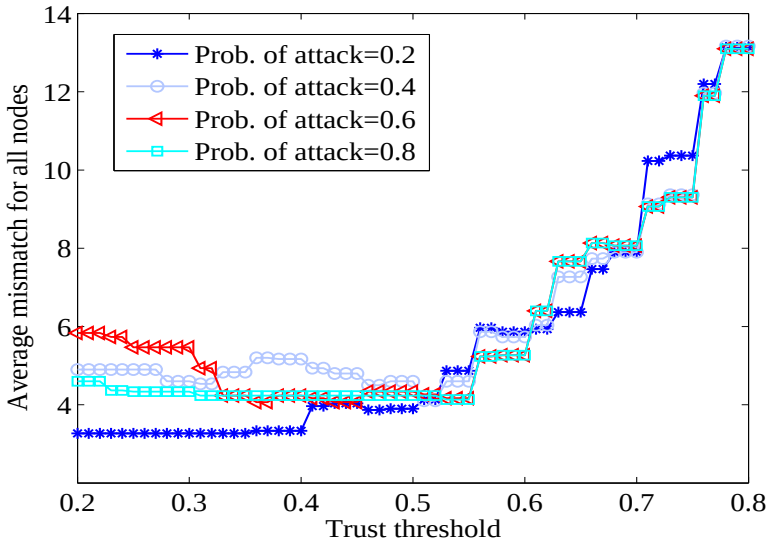
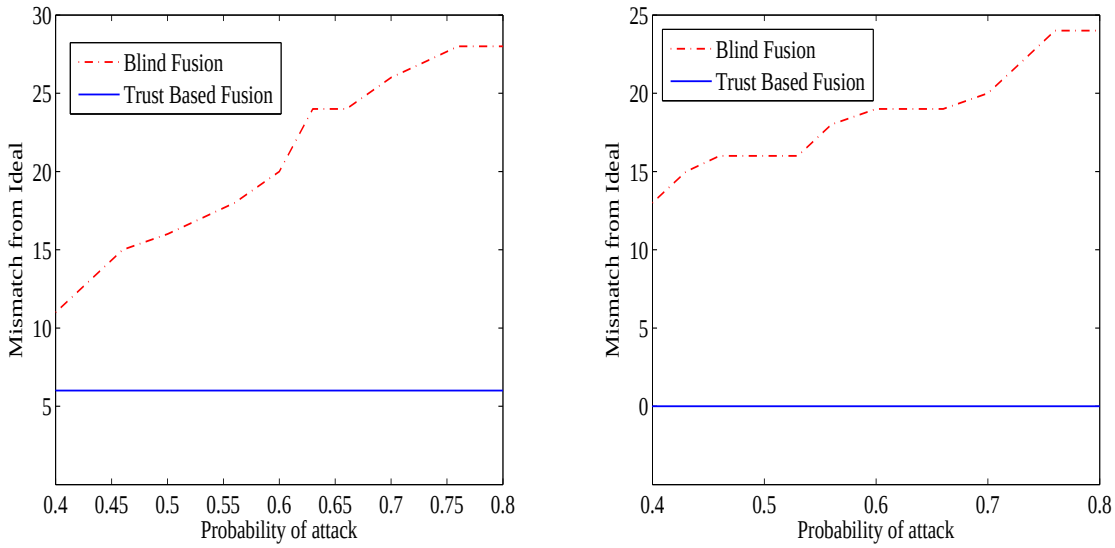


Figure 8.21: Optimal Threshold selection for trust based fusion

8.5.2 Fusion results for transient state: Individual node perspective

Using $\Gamma_{opt} = 0.50$, from Fig. 8.22(a) and Fig. 8.22(b), it is evident that by using trust based fusion the nodes 10 and 23 have mismatches significantly lower than if they had blindly fused occupancy data without trust filtration. To measure what percentage of nodes are benefited using our framework of TBF, even at high probabilities of attack of 0.8, 90% of the nodes have average mismatches less than 6 (15%) shown in Fig. 8.23. The results reflect the effectiveness of trust based fusion over blind fusion.



(a) (b)

Figure 8.22: Trust based fusion for optimistic trust model with $\Gamma_{opt} = 0.5$: An individual node snapshot

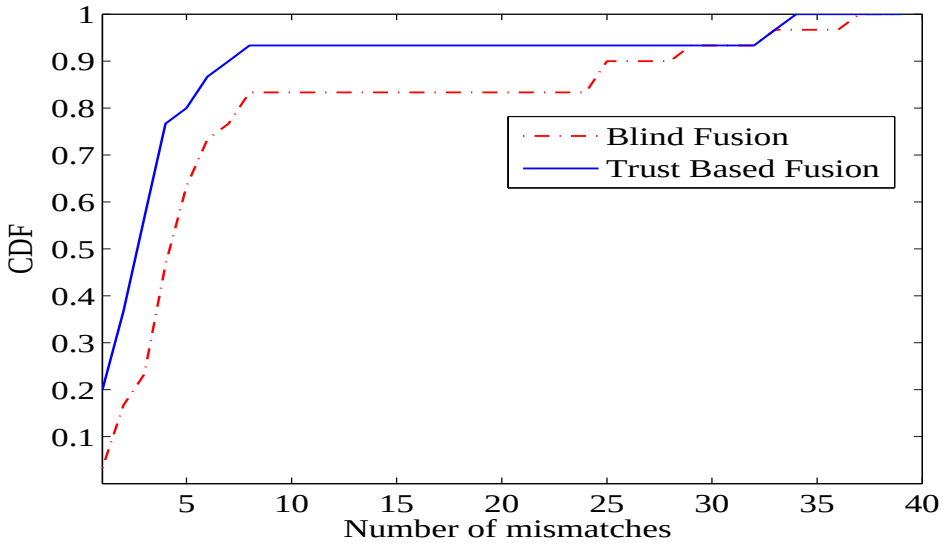


Figure 8.23: CDF of mismatches

8.5.3 Fusion results for the steady state: Overall network perspective

In Fig. 8.24 and Fig. 8.25, we observe that the percentage of mismatches are far less for trust based fusion which filters out potentially dishonest nodes rather than blind fusion. This is true for both attack strategies. For lower magnitudes of attack the network is not significantly damaged hence they are allowed, and hence the blind fusion and trust based fusion have almost similar mismatches. This is because we do not want to lose the cooperation that dishonest nodes contribute when the attack magnitude is low. However, when the attack magnitude increases, the trust values of malicious nodes fall below the desired threshold Γ_{opt} and their false opinions get filtered out decreasing mismatches. The results from this are indicative of the effectiveness of the proposed trust based fusion.

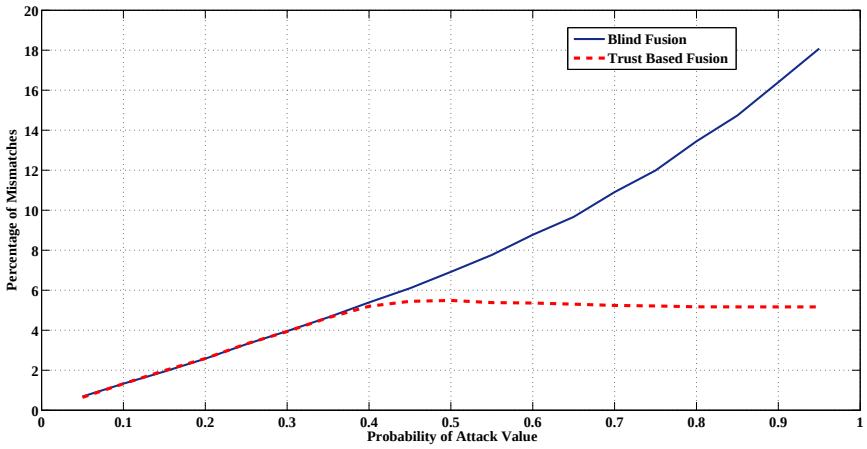


Figure 8.24: Performance comparison of trust based fusion under P_{attack}: Steady state

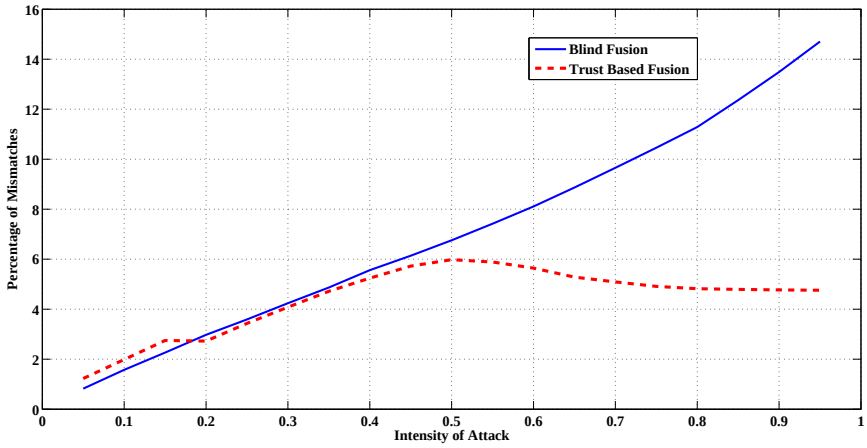


Figure 8.25: Performance comparison of trust based fusion under I_{attack}: Steady state

8.6 Inversion based Fusion: Utilizing Misleading Information

We simulate 100 randomly scattered nodes out of which 30% nodes are programmed to be malicious. All nodes continuously scan 40 channels, record the signal power on each of them, and create the binary occupancy vector which they then advertise. The malicious nodes attack (i.e., change the bits in the channel occupancy vector) with a probability between 0.5 to 1.0. It is to be noted that lower attack probability do not significantly affect the network.

8.6.1 Log weight measurement

In Fig. 8.26, we see the difference between the average weights for all honest nodes and the average weight for all malicious nodes. We see a significant difference between the two sets, with honest node line well over 0 and malicious nodes below 0. As discussed earlier we see

the malicious node's weight on the negative y-axis and the weights of honest node on the positive y-axis. We run the simulation for different P_{attack}, and observe that more aggressive they attack, their weights and trust decrease. The observation is also true for all malicious nodes as well as any individual malicious nodes as shown in Fig. 8.27.

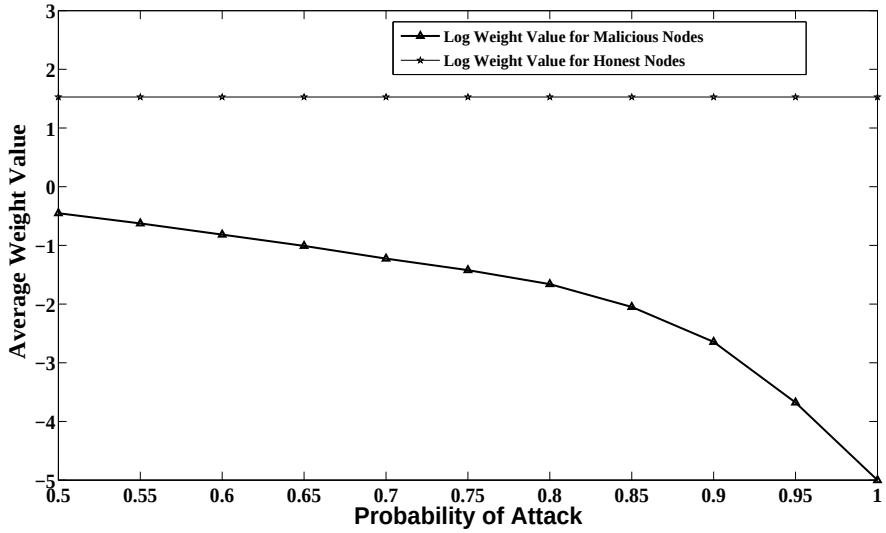


Figure 8.26: Log weighted trust between honest and malicious nodes

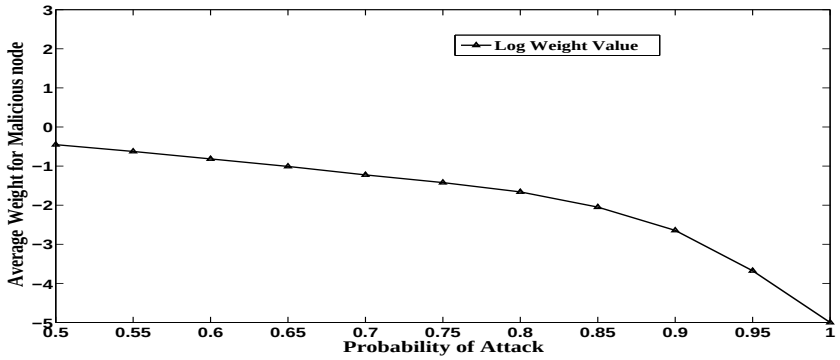


Figure 8.27: Log weighted trust over P_{attack}

8.6.2 Optimal threshold for inversion based fusion schemas

In trust based fusion, we used a trust threshold of 0.50, where nodes below that value were considered malicious and their occupancy vector was discarded from fusion. From Fig. 8.28, it is evident that the minimum possible mismatch fraction is achieved at 0.0 for all probabilities of attack. As we lower the threshold, more and more malicious nodes will be included in no inversion, hence the higher number of mismatches. We show the results for mismatch fraction over different P_{attack} considering different candidate weight thresholds in Fig. 8.29 and observe that the lowest mismatch fraction for all P_{attack} is the one that corresponds to weight threshold of 0. From the above two observations, we can infer that for all P_{attack}, W_{opt} should be 0.

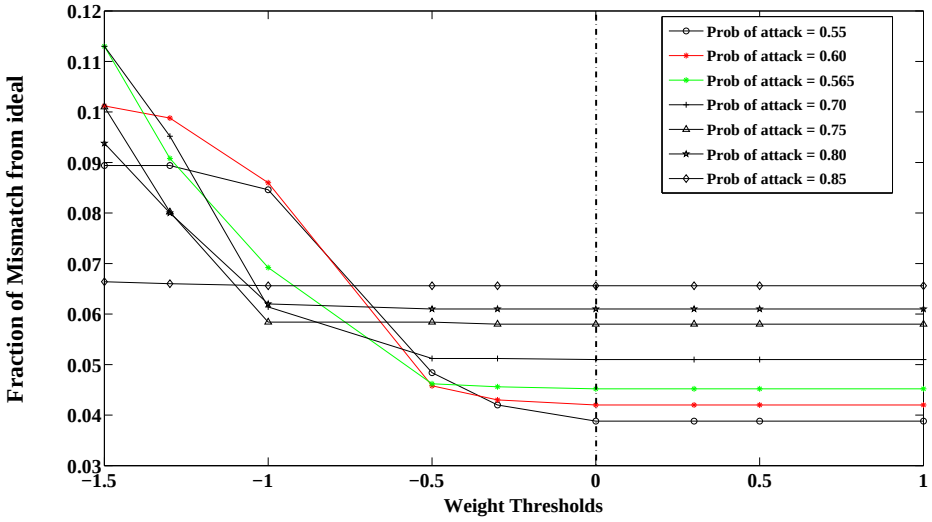


Figure 8.28: Optimal threshold for invoking inversion based fusion

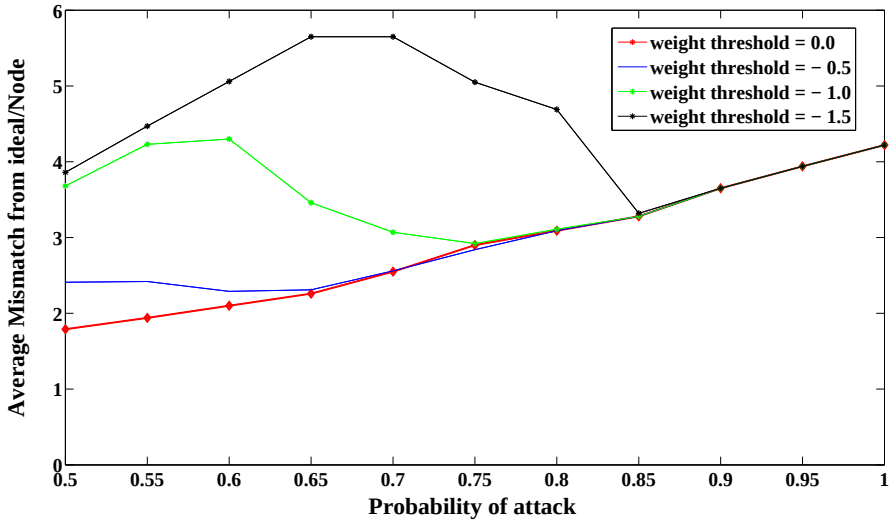


Figure 8.29: Percentage of mismatches for different candidate thresholds

8.6.3 Selective inversion and complete inversion

The nodes whose weights are below or equal to 0 are considered potentially malicious node and they are candidates for inversion schemes. Using the entire range for practical P_{attack} values for selective and complete inversions, we found that for lower P_{attack} selective inversion performs better as evident from Fig. 8.30. As P_{attack} takes higher values, the undecided channels (i.e., neither matches nor mismatches) are not taken into account in selective inversion and hence the mismatches increase. However, for the complete inversion the reverse happens. As P_{attack} increases, more channels are inverted and an inverted vector of a malicious node is closer to the actual occupancy. This is due to the gain in cooperation from even the malicious nodes that we harness. The point where the two inversion schemes SI and CI cross is termed as 'crossover point'. Before the crossover point, selective inversion works

better and after the crossover point complete inversion works better. From simulations, we find that the crossover point occurs at $P_{attack} = 0.65$ as justified and shown in Fig. 8.31. With the crossover point known, we are able to back calculate the threshold W_{min}. Recall, the nodes are not aware of the probability of attack launched by malicious nodes but their weights are a reflection of the attack probability.

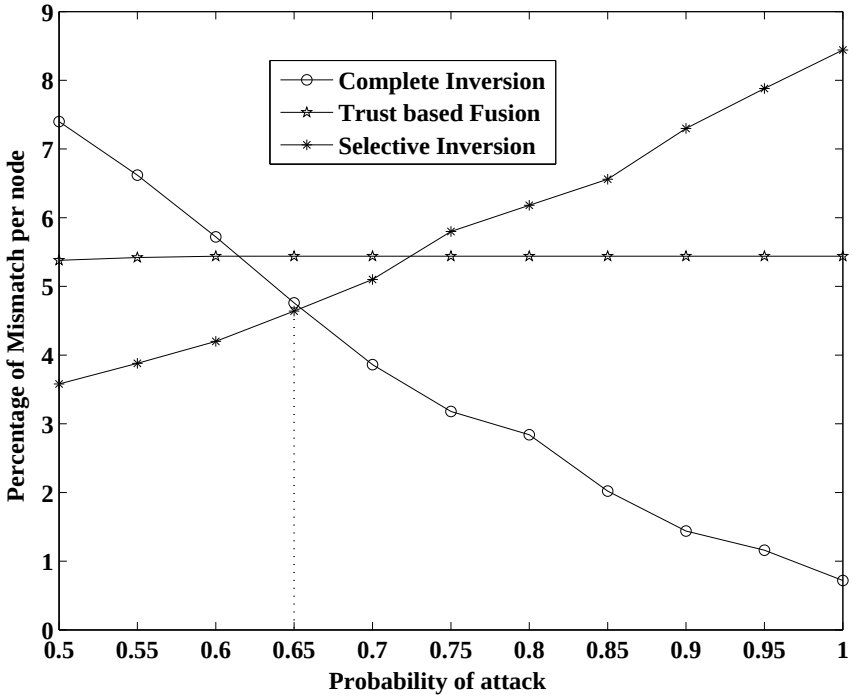


Figure 8.30: Comparative study of proposed fusion schemes with Trust based Fusion; Percentage of mismatches for all P_{attack}

8.6.4 Threshold selection for CI and SI fusion: A combined approach

In order to examine the nature of the crossover point, we consider different densities of malicious nodes (P_{mal}) for both the fusion schemes and see whether there is a consensus on

the crossover point. From Fig. 8.31 confirms that $P_{attack} = 0.65$ is the cross-over point for both inversion schemes for $\rho_{mal} = 0.2, 0.3$ and 0.4. Though it is obvious to use an inversion scheme based on the probability of attack, the problem is that the regular nodes would not know the probability of attack. However, using the log-weight based trust evaluation, they can compute the weights, w_i, which indirectly captures the probability of attack. Thus, there is a one-to-one correspondence between p_{attack} and w_i^{mal}.

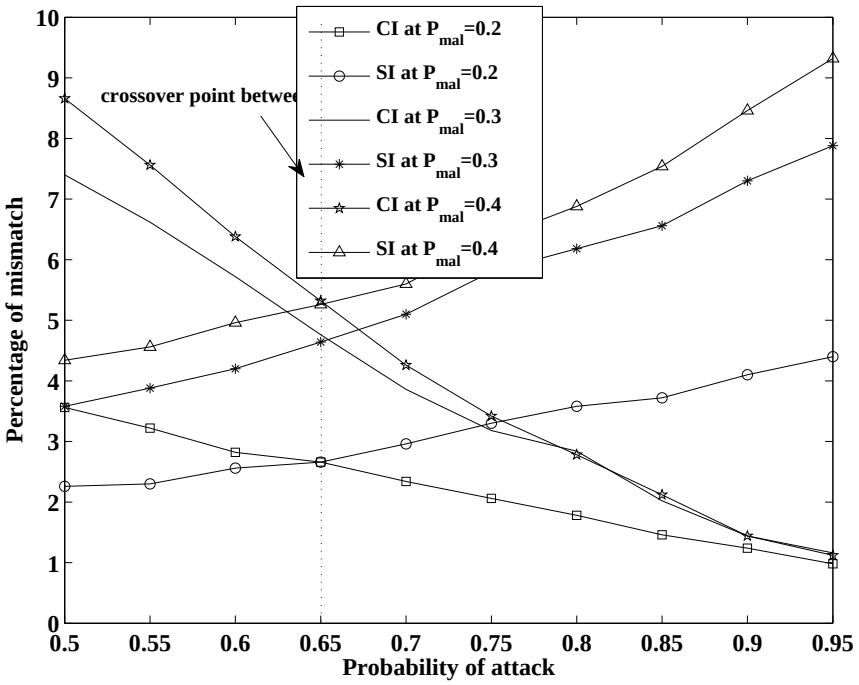


Figure 8.31: Crossover point for different malicious node densities

Table 8.1: Average Trust at $P_{crossover} = 0.65$ for different ρ_{mal}

Malicious Node Density	Average Weight Value for $P_{attack} = 0.65$
0.2	-1.023909
0.3	-1.009977
0.4	-1.037043

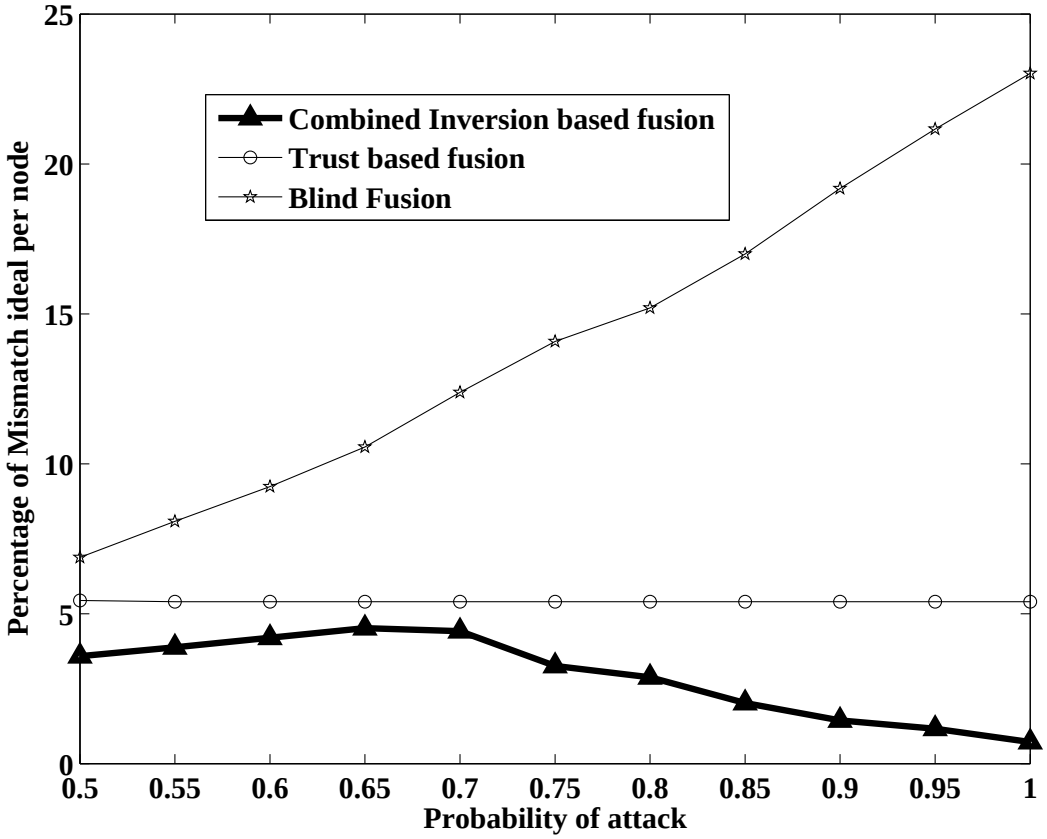


Figure 8.32: Combined inversion compared with Trust based and Blind Fusion

We find the average W_i for malicious nodes that corresponds to $P_{attack} = 0.65$ for each value of ρ_{mal} (from Table 8.1) and call it $W_{crossover}$. Interestingly, the weights for malicious nodes at $P_{crossover}$ is almost the same. Hence we make $W_{crossover} = W_{min} = -1.00$ the threshold which decides which inversion scheme is to be invoked. Knowing W_{min} and noting that P_{attack} and W_i are inversely related, we simply use selective inversion for $W_i > W_{min}$ and complete inversion for $W_i < W_{min}$. The result of the combined inversion fusion is compared with blind fusion and trust based fusion in Fig. 8.32. Either way from the plot, it is apparent

that an isolation based trusted based fusion is always the second best, and the combined inversion based inclusive technique works better than the isolative trust based technique.

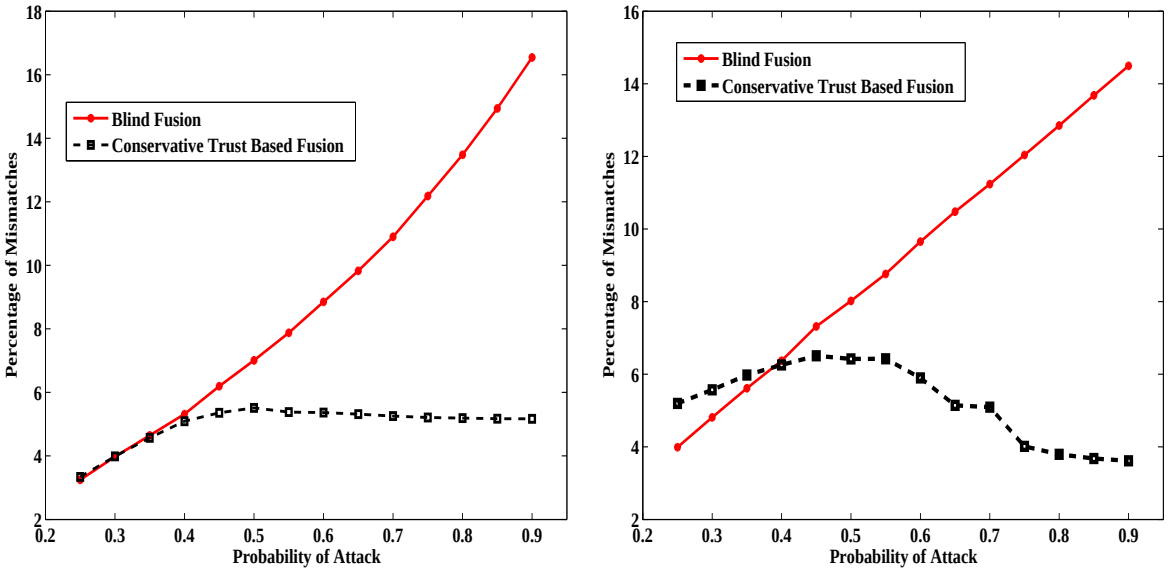
8.7 Trust based Fusion: Conservative Trust Model

For the fusion results, we use a trust threshold of 0.0 against instantaneous conservative trust weights, to discard the information from such nodes. This approach uses a different threshold than what was used for malicious node identification, because on each time slot the actual realization of P_{attack} may be different from the mean value and does not reflect the degree of falsification on a particular time slot. We use the trust threshold of 0, by the same logic we chose 0.50 as threshold, when trust was bounded between 0 and 1.

8.7.1 Robust fusion results for conservative trust weights

Fig. 8.33(a), we show the performance of the proposed conservative trust based fusion model as opposed to blind fusion, and report far less percentage of mismatches. The same also holds true if the malicious nodes launch collaborative attacks where they agree upon select channels. However, when P_{attack} is low in case of collaborative attack, the trusted fusion gives more mismatches than blind fusion as can be observed in Fig. 8.33(b). Hence an important conclusion is that when P_{attack} is low ($<< 0.5$), and strategy is collaborative SSDF, trust based fusion that tries to disregard nodes based on their trust values in not an

effective approach. This is effectively the case of selfish SSDF attacks, where magnitude of attack is low and nodes have interest on specific channels agree upon falsification on those channels. This gives motivation for a separate defense strategy against such nodes who employ collaborative SSDF attack on a minority number of channels of interest. We keep this open problem for our future work.



(a) (b)

Figure 8.33: Performance of conservative trust based fusion:(a) Under non-collaborative attack (b) Under collaborative attack

8.7.2 Comparison of conservative robust fusion with existing research

In Fig. 8.34, we show that under high density of collaborative malicious nodes, majority voting based node reputation discussed in [21], cannot defend against errors induced by

such attacks. On the other hand our proposed trust based fusion can preserve the integrity of cooperative sensing results in a better way.

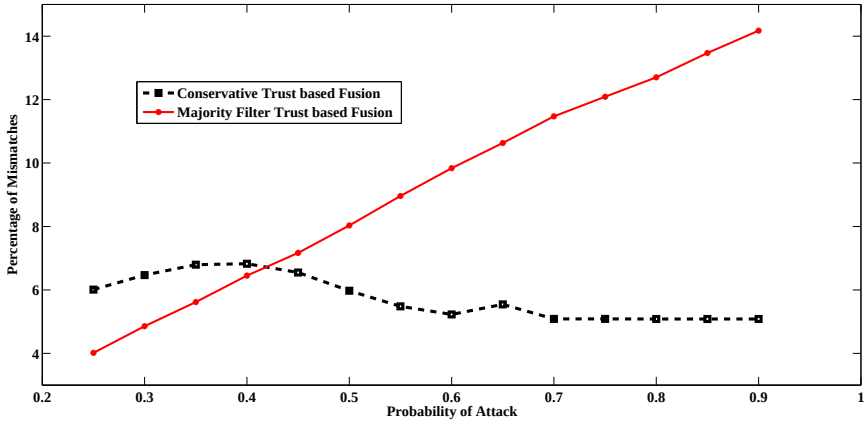


Figure 8.34: Comparing CTBF with majority voting based exclusion for high density collaborative attack

8.8 Trust based Channel Preference Under Selfish Collaboration

To validate the trust model and to compute the net trust for each channel, we simulate a distributed network over an area of 3000 × 3000 m with 20 primaries; thus 20 channels. We consider a total of 30 (honest and malicious) mobile secondary nodes. The received power at each receiver depends on the path loss factor ω which was varied between 3 and 5. We show the results for different nodes that belong to different parts of the network. As for the update periodicity, we update the beliefs every 3 time slots which we refer to as a *window*. The sensitivity factor λ was kept at 0.2.

8.8.1 Trust based channel preference

We consider 40% of the nodes as selfish which collaboratively attacked the channels $S = \{1, 3, 5, 7, 9, 11, 15, 16, 19\}$. The channels attacked remained static over time. In Fig. 8.35, we show the net trust of the channels sorted in an descending manner as calculated by an honest node 12.

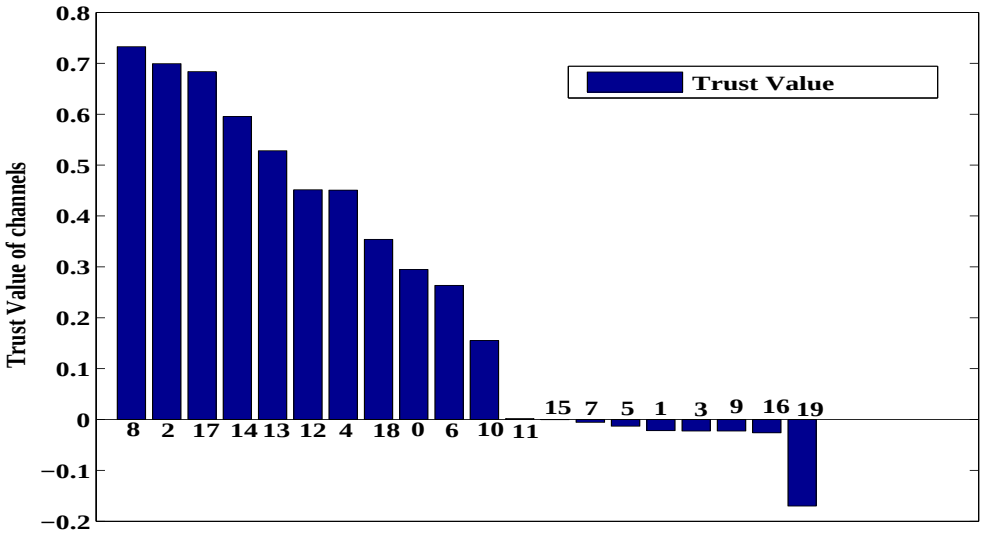


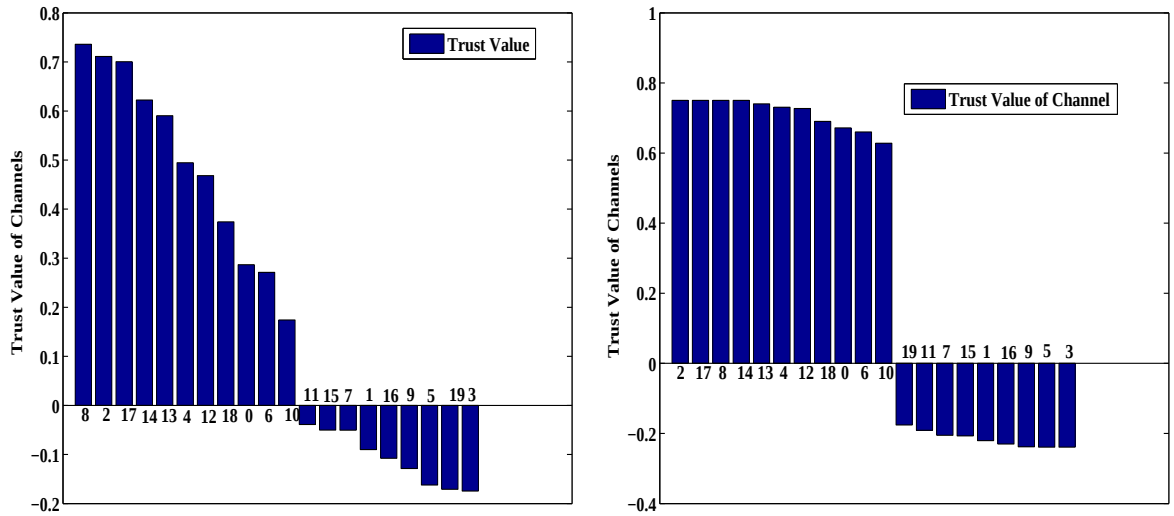
Figure 8.35: Channel preference order by node 12 when $\omega = 3.5$

As expected, the 9 channels that were attacked have significantly lower trust values than others. For node 12, the best channel is 8 and the second best is 2. The worst channel is 19. Therefore, the fused decision on channel 8 can be trusted the most followed by 2 and so on. The strategy for node 12 will be to check whether channel 8 can be used to communicate with its intended receiver. If not, it will check channel 2 and so on. Our results show that for both variants (i.e., static and dynamic) the channels that were attacked have a significantly lower trust value than channels that were not. Moreover, for static attacks it is also possible

to identify the channels that were selected by the adversarial group. We also show that the proposed trust model is effective even for a high density of selfish nodes. We demonstrate that the proposed accuracy of channel centric defense mechanism is preserved under node mobility and various pathloss environments.

8.8.2 Channel preference with more selfish nodes

Now, we set the number of selfish nodes to 60% of the total number of nodes i.e., 18. In Fig. 8.36(a), we show the net trust of the channels sorted in an descending manner as calculated by the honest node 14. As per most previous works, it is difficult to distinguish dishonest behavior when the adversary density is higher than 50% [36].



(a) (b)

Figure 8.36: Channel preference of node no. 14 with 18 selfish nodes: (a) For $\omega = 3.5$ (b) For $\omega = 5.0$

It is to be noted that node 14 is able to identify the 9 channels that were attacked inspite of the increased number of selfish nodes. The trust values for those channels are even lower– enabling a better differentiation of the channels. However, the order is different from what was observed by node 12 because of being at a different location. Fig. 8.36(b), shows that results hold true for a higher value of $\omega = 5.0$.

8.8.3 Trust propagation under static selfish attacks

In Fig. 8.37, we observe how trust values evolve over time with the proposed trust update model for static attacks. It clearly exhibits the difference between a channel that was attacked and a channel that was not. We consider two channels: one from the attacked channel set (i.e., 1) and another from the non-attacked channel set (i.e., 4). We plot the *weighted moving average* of the net trust after every 3 time slots. As time progresses, we see a clear difference in the trust values for channels 1 and 4.

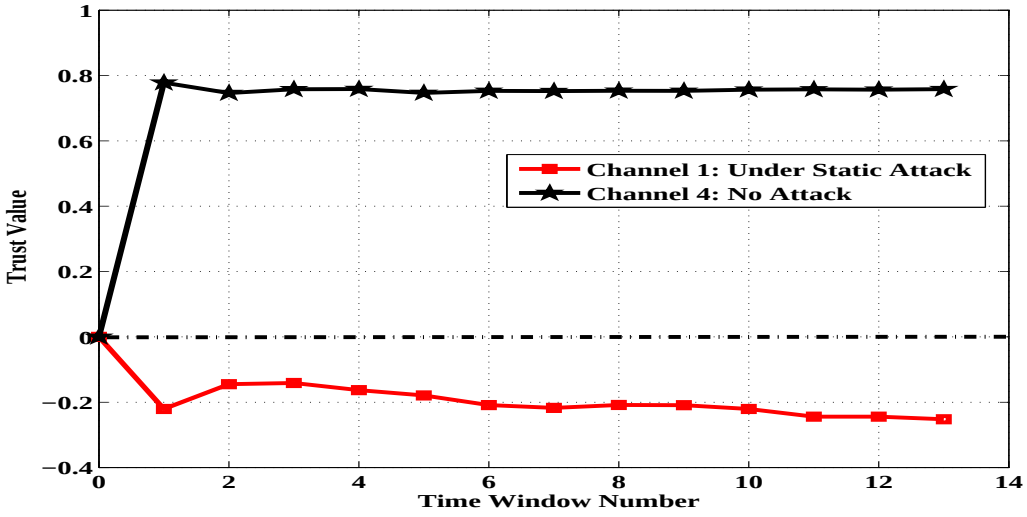


Figure 8.37: Net trust for attacked and not-attacked channels

8.8.4 Effect of fraction of selfish nodes

We investigate how the trust values for channels vary with different fractions of selfish nodes. In Fig. 8.38, we show the average trust values for attacked channel set and the non-attacked channel set, for increasing number of attackers. As expected, there is hardly any change on channels that are not attacked, whereas, the average trust value for the channels that were attacked decreases. This result is in contrast with the commonly used voting schemes or Kullback-Leibler divergence based defense models that fail to detect anomaly when the number of attackers is more than 50%.

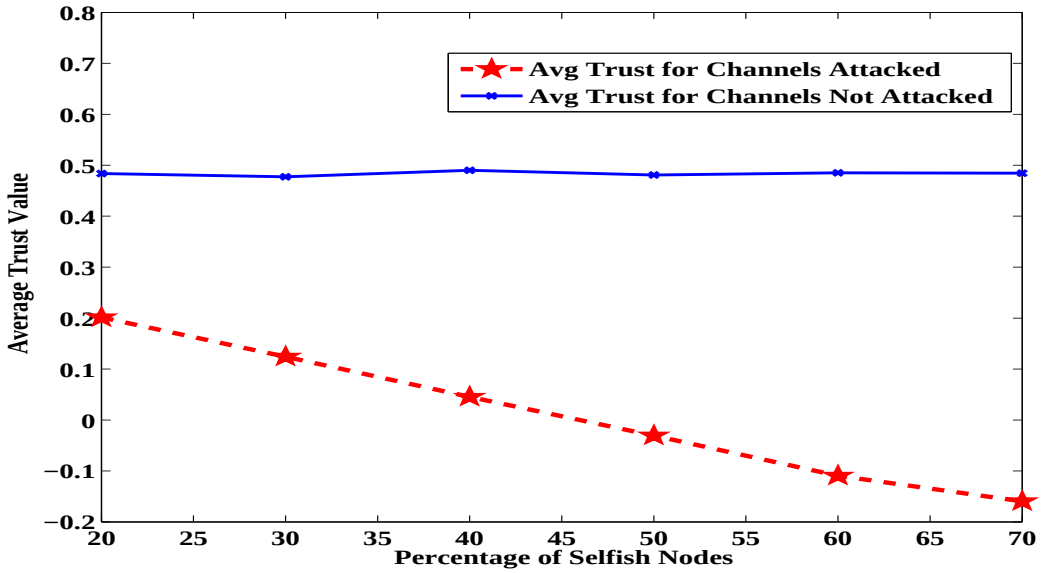
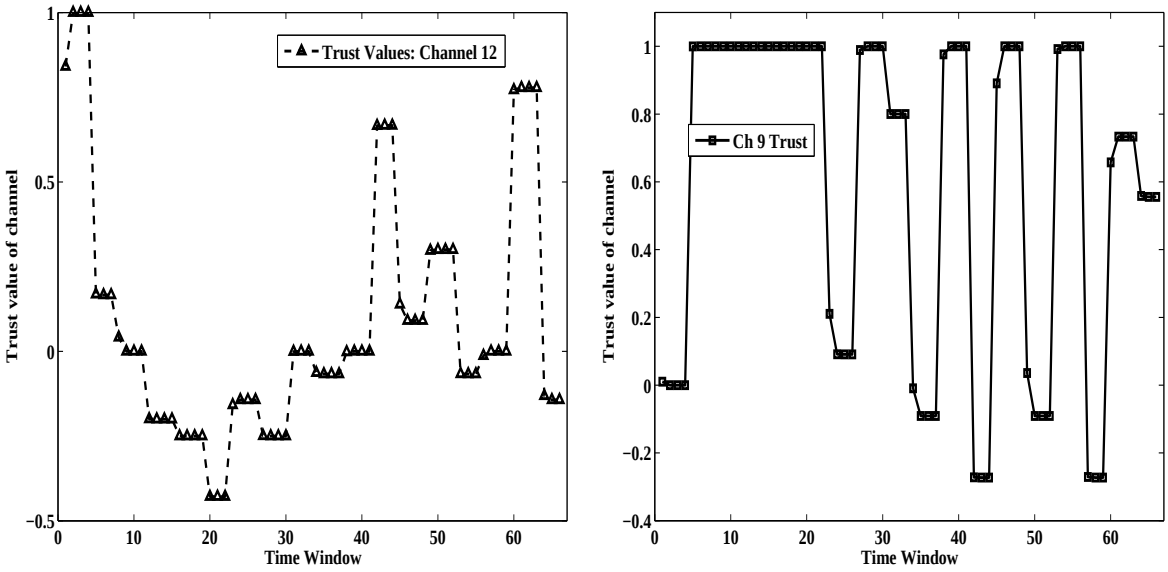


Figure 8.38: Average trust with increasing number of selfish nodes

8.8.5 Trust variation under dynamic selfish attacks

To prove that our model is able to capture the frequent changes in the attacked channel set, we vary the attacked channel set every 10 time slots on an average. We show how the net trust is updated for two specific channels (12 and 9) in Fig. 8.39(a) and Fig. 8.39(b) respectively.



(a) (b)

Figure 8.39: Net trust updates for dynamic attacks (a) Channel 12 (b) Channel 9

Note, the plots are shown in time windows (3 time slots = 1 window). These two representative channels (9 and 12) were intermittently and randomly chosen by the coalition of selfish nodes. Trust of channel 12 starts with a high value for 10 time windows, but then attacked consistently for next 15 windows (i.e., 45 time slots); hence its trust exhibits a steady decrease. After that channel 12, is intermittently chosen for attack, which causes the rapid

fluctuations of the trusts values. Channel 9 is not attacked for 22 windows then attacked for a few windows. Hence, its trust value is initially high followed by a sudden decrease which stays unchanged for 4 windows and then increases for the remaining windows.

8.9 Generlized Bayesian Framework for Quantifying Decision Reliability

We simulate a generic centralized system with 100 nodes. Inputs from all nodes are monitored by an imperfect monitoring mechanism that produces three possible outcomes. The probability of detection, P_{det}, is varied to capture its effects on decision reliability.

An adversary attacks and compromises different sets of inputs over time. The number of inputs compromised vary over time slot; although the long-term average of the number of inputs compromised, denoted by P_a, remains the same. We study the decision reliability for different values of P_a, and plot instantaneous and moving average of decision reliability. We first present the results for an optimistic system and then for a more conservative system.

8.9.1 Optimistic decision reliability: Instantaneous and average

In Fig. 8.40, we plot both the instantaneous and average decision reliability for the optimistic reliability model when the adversary launches attacks with $P_a = 0.5$. We observe that the decision reliability fluctuates over time. As expected, with sufficient observations, the moving

average of decision reliability converges to a steady state reliability equal to $1 - P_a$ which in this case is 0.5.

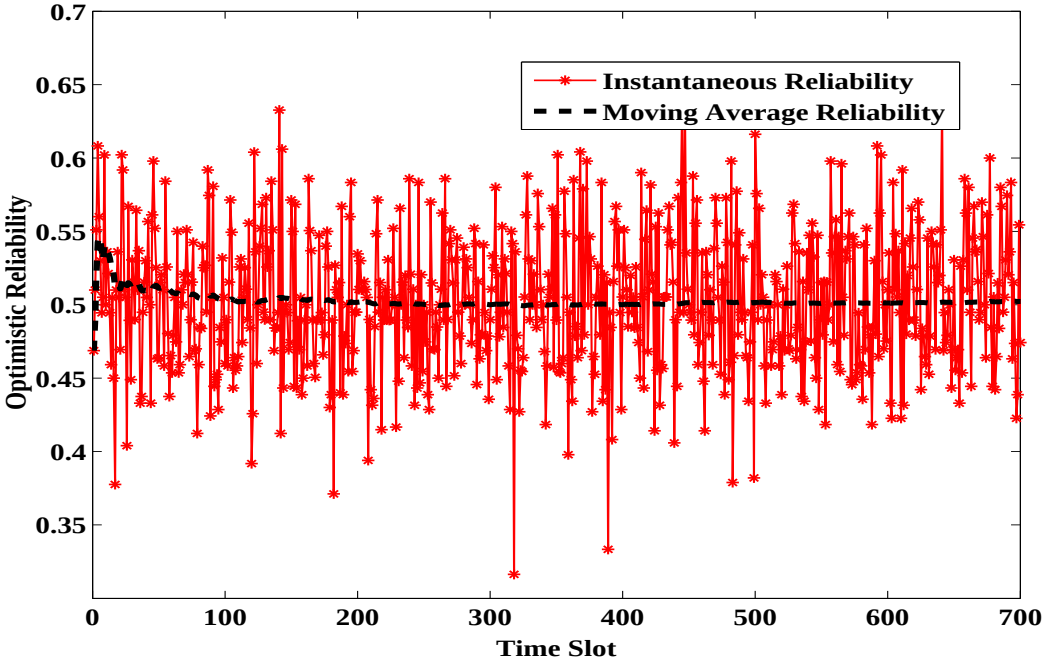


Figure 8.40: Instantaneous and average decision reliability with $P_a = 0.50$ and $P_{det} = 0.8$

8.9.2 Decision reliability and entropy

In Fig. 8.41, we plot steady state decision reliability with increasing P_a and for two different values of P_{det}. The plots show a steady decrease of decision reliability values for P_{det} 0.5 and 0.9. Recall, with different P_{det} the values of average decision reliability may differ but the relation between R_s^o and P_a does not change with change in P_{det} unlike R_s^c. This is because inputs chosen by the adversary is uniformly random. The conservative reliability R_s^c, also falls linearly with increasing P_a, but the slope or rate of this change varies for the different

values of P_{det}. This is because the conservative model does not account for the undecided ones. Hence a lower value of P_{det} yields a lower reliability value than a higher value of P_{det}.

Furthermore, Fig. 8.42 shows the entropy values of the system under the same set of P_{det}. This shows that a system with higher P_{det} has a lower uncertainty on the decision reliability estimate for all values of P_a. This is useful when we use optimistic system reliability which uses a fraction of undecided ones to contribute to the final reliability value.

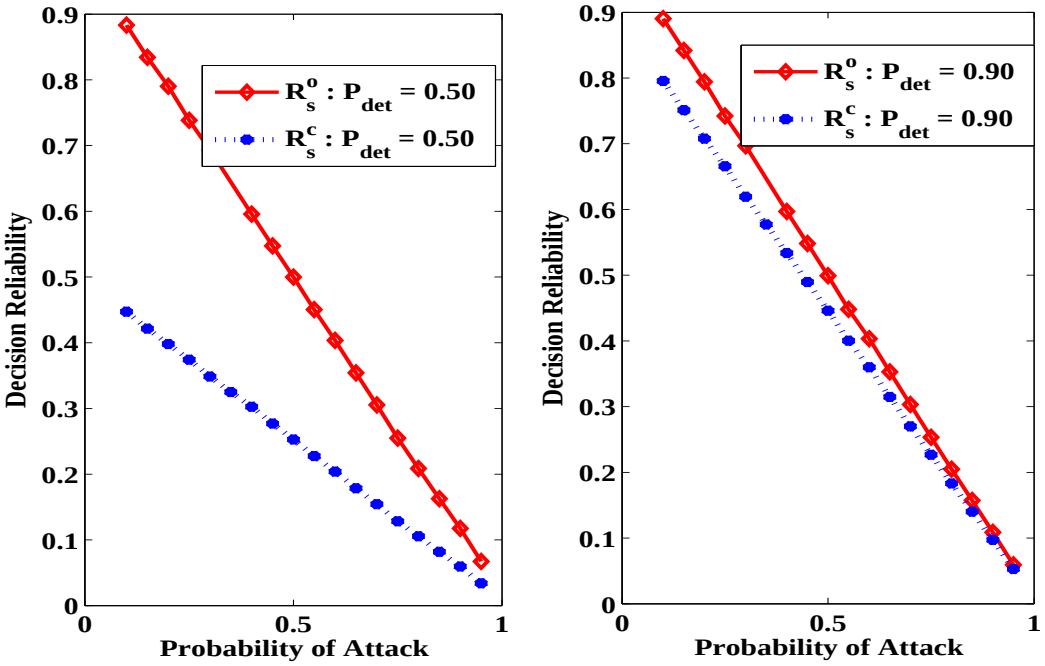


Figure 8.41: Optimistic and Conservative Reliability over P_a for different P_{det}

In Fig. 8.43, we show how entropy changes with increasing probability of attack. As time evolves and P_{det} improves, the uncertainty (i.e., entropy) associated with the decision reliability decreases indicating increased confidence on the reliability values.

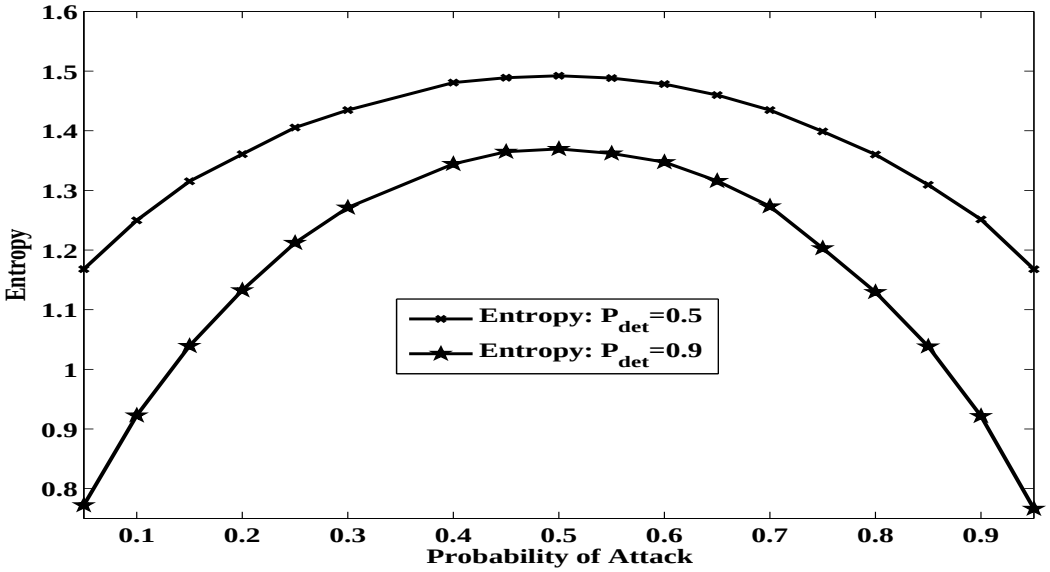


Figure 8.42: Entropy over different values of P_a

8.9.3 Decision reliability with time variant P_a

Some adversaries may choose to attack short-term with low or high attack probability; however, maintaining the long-term average value of P_a. For example, an adversary attacks less initially conserving its attack resources for future and eventually attacking more (under favorable conditions). In Fig. 8.44, we investigate how the proposed model behaves in such cases. We consider an adversary with $P_a = 0.5$. The first 500 slots are attacked less and next 500 slots are compensated by attacking more. We observe that decision reliability progressively moves towards the expected reliability, although no strict convergence on the decision reliability is achieved.

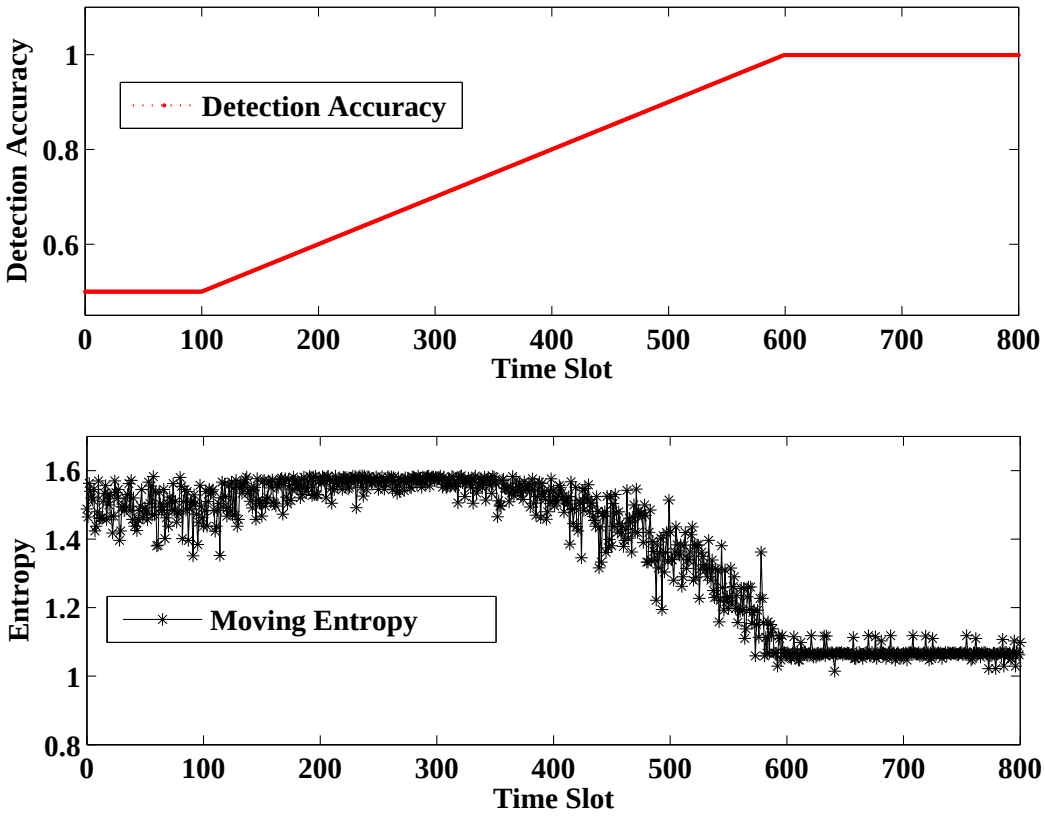


Figure 8.43: Conservative decision entropy with incremental increase of P_{det}

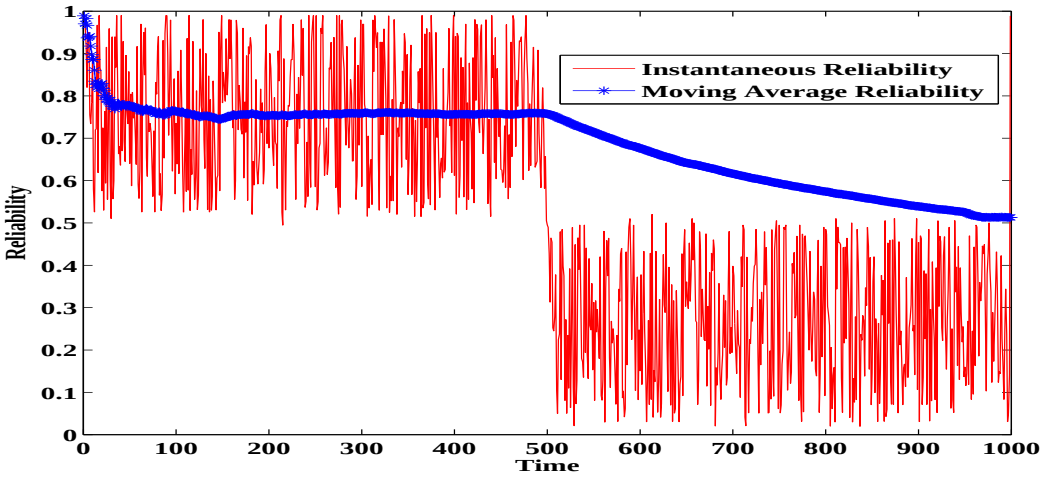


Figure 8.44: Decision Reliability under Non-Uniformly distributed $P_a = 0.50$

8.9.4 Conservative reliability model

In Fig. 8.45, we plot the changes in conservative decision reliability, over time with gradual increase in P_{det}. As mentioned earlier, if the system evolves into more accurate monitoring, the decision reliability also improves, although $R_s \neq 1 - P_a$. Thus, having a conservative decision reliability is not unfair, given that risk associated with it is high, and there is scope for improving reliability when detection accuracy increases.

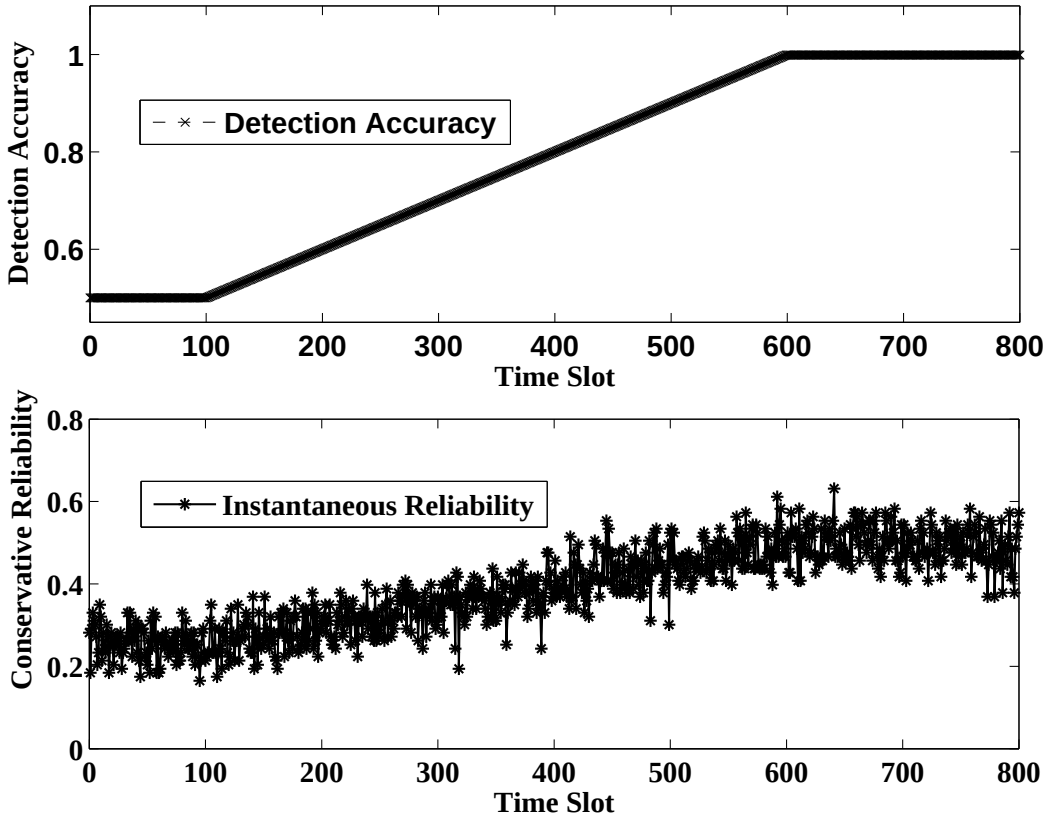


Figure 8.45: Conservative decision reliability with increasing P_{det}

Through simulation experiments, we showed how the decision reliability changes with attack probability which also affects the detection accuracy of the underlying monitoring

mechanism. Using entropy, we provided a way to evaluate the certainty of the reliability calculation. Finally, we showed how the decision reliability decreases when the attack probability increases.

CHAPTER 9: CONCLUSIONS

This dissertation addressed the vulnerabilities of cooperative spectrum sensing in DSA networks and propoased techniques to mitigate them. We show how malicious and selfish nodes can launch a variety of attacks. In that regard, we discussed various attack models including probabilistic multi-channel SSDF, collaborative SSDF, on-off SSDF attacks and their variants.

In order to quantify trustworthiness of nodes, we proposed an anomaly monitoring scheme is effective in capturing anomalies using mismatches while also accommodating uncertainty because a honest node may be legitimately outside the reception range of primary. Uisng the invert sequence the proposed scheme, we show how we can utilize misleading information under certain conditions by intelligent flipping of data shared. We propse an optimistic trust model quantifying trust for a low risk system which uses instantaneous trust values while fusing spectrum occupancy data from neighbors. We provide a way to rank nodes based on their trustworthiness and disregard reports from nodes whose trust values fall below a threshold. While instantaneous trust values are useful for robust fusion from an individual node's perspective, long-term average trustworthiness for a node maintained and updated over time as seen by all neighbors of a node (known as reputation) is useful for network administrative tasks like isolating nodes from a network or taking other pros-

ecutable actions. We observe significant improvement in the performance of trust based fusion over blind majority fusion for all practical values of magnitude of attack and for both collaborative and non-collaborative attacks. Then we propose a conservative trust model that has the ability to cope with strict risk attitude, variation of evidence due to varying pathloss environment, different levels of aggression and uncertainties in the monitoring evidence. Using support vector machines over several training sets, we could classify malicious nodes more accurately than some existing methods that use pairwise entropy or majority voting based exclusion. The conservative trust based fusion also improves accuracy in a similar way as the optimistic trust based fusion. The proposed asymmetric moving average model is able to deal with on-off attacks; however, sometimes honest reports may be altered due to noise. Our results from asymmetric moving average for trust maintenance shows we can differentiate between on-off attacks and noise. We observe that malicious nodes take longer to improve their trust than nodes experiencing noise even when malicious nodes show good behavior for a very long time. Apart from node centric approaches we argue the need for a channel centric approach towards identifying the trustworthiness of fused decision on each channel. In this regard, we propose a channel centric Bayesian inference framework that helps rank channels. We also show that the proposed framework is generic enough to model decision reliability in any cooperative decision making system.

Overall, this dissertation improves upon and extends our understanding on the vulnerabilities of DSA networks. This is the first work to investigate multi-channel collaborative

SSDF attacks and provide novel strategies to defend them. This study provides key insights on the design of DSA networks with operational assurance under adversarial situations.

LIST OF REFERENCES

[1] F. communications commission (FCC), "Spectrum policy task force," Nov 2002. Rep. ET Docket no. 02-135.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 201–220, Feb 2005.

[3] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio:PhD thesis, KTH stockholm," June 2000.

[4] M. Buddhikot and K. Ryan, "Spectrum management in coordinated dynamic spectrum access based cellular networks," in *New Frontiers in Dynamic Spectrum Access Networks, (DySPAN), First IEEE International Symposium on*, pp. 299–307, Nov 2005.

[5] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Signals, Systems and Computers, Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 1, pp. 772–776 Vol.1, Nov 2004.

[6] A. Ghasemi and E. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *New Frontiers in Dynamic Spectrum Access Networks, (DySPAN), First IEEE International Symposium on*, pp. 131–136, Nov 2005.

[7] S. Bhattacharjee, S. Sengupta, and M. Chatterjee, "Vulnerabilities in cognitive radio networks: A survey," *Computer Communications*, vol. 36, no. 13, pp. 1387 – 1398, 2013.

[8] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM, The 27th Conference on Computer Communications. IEEE*, pp. 13–18, April 2008.

[9] S. Bhattacharjee, S. Debroy, and M. Chatterjee, "Trust computation through anomaly monitoring in distributed cognitive radio networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), IEEE 22nd International Symposium on*, pp. 593–597, Sept 2011.

[10] S. Bhattacharjee, S. Debroy, M. Chatterjee, and K. Kwiat, "Trust based fusion over noisy channels through anomaly detection in cognitive radio networks," in *Proceedings of the 4th International Conference on Security of Information and Networks*, SIN, pp. 73–80, ACM, 2011.

[11] S. Bhattacharjee, S. Debroy, M. Chatterjee, and K. Kwiat, "Utilizing misleading information for cooperative spectrum sensing in cognitive radio networks," in *Communications (ICC), IEEE International Conference on*, pp. 2612–2616, June 2013.

[12] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, pp. 2127–2159, Sept. 2006.

[13] J. Mitola and J. Maguire, G.Q., "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, pp. 13–18, Aug 1999.

[14] F. Digham, M.-S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels," *Communications, IEEE Transactions on*, vol. 55, pp. 21–24, Jan 2007.

[15] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," in *New Frontiers in Dynamic Spectrum Access Networks, (DySPAN) First IEEE International Symposium on*, pp. 137–143, Nov 2005.

[16] S. Mishra, A. Sahai, and R. Brodersen, "Cooperative sensing among cognitive radios," in *Communications, (ICC) IEEE International Conference on*, vol. 4, pp. 1658–1663, June 2006.

[17] C. Sun, W. Zhang, and K. Letaief, "Cooperative spectrum sensing for cognitive radios under bandwidth constraints," in *Wireless Communications and Networking Conference, (WCNC) IEEE*, pp. 1–5, March 2007.

[18] C. Sun, W. Zhang, and K. Ben, "Cluster-based cooperative spectrum sensing in cognitive radio systems," in *Communications, (ICC) IEEE International Conference on*, pp. 2511–2515, June 2007.

[19] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *INFOCOM, Proceedings IEEE*, pp. 729–737, March 2012.

[20] T. Clancy and N. Goergen, "Security in cognitive radio networks: Threats and mitigation," in *Cognitive Radio Oriented Wireless Networks and Communications, (CrownCom) 3rd International Conference on*, pp. 1–8, May 2008.

[21] P. Anand, A. Rawat, H. Chen, and P. Varshney, "Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks," in *Communication Systems and Networks (COMSNETS), Second International Conference on*, pp. 1–9, Jan 2010.

[22] D. Gambetta, "Can we trust trust?," in *Trust: Making and Breaking Cooperative Relations*, pp. 213–237, Basil Blackwell, 1988.

[23] T. Grandison and M. Sloman, "Trust management tools for internet applications," in *Proceedings of the 1st International Conference on Trust Management*, iTrust, pp. 91–107, Springer-Verlag, 2003.

[24] L. Mui, M. Mohtashemi, and A. Halberstadt, "A computational model of trust and reputation for e-businesses," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, HICSS, p. 188, IEEE Computer Society, 2002.

[25] S. P. Marsh, "Formalising trust as a computational concept," 1994.

[26] A. Jsang, "Artificial reasoning with subjective logic," 1997.

[27] A. Josang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618 – 644, 2007. Emerging Issues in Collaborative Commerce.

[28] W. Wang, H. Li, Y. Sun, and Z. Han, "Catchit: Detect malicious nodes in collaborative spectrum sensing," in *Proceedings of the 28th IEEE Conference on Global Telecommunications*, GLOBECOM, pp. 5071–5076, IEEE Press, 2009.

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

[30] P. K. Varshney, *Distributed Detection and Data Fusion*. Springer-Verlag New York, Inc., 1st ed., 1996.

[31] A. Wald, "Sequential tests of statistical hypotheses," *The Annals of Mathematical Statistics*, vol. 16, no. 2, pp. pp. 117–186, 1945.

[32] Z. Jin, S. Anand, and K. P. Subbalakshmi, "Mitigating primary user emulation attacks in dynamic spectrum access networks using hypothesis testing," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, pp. 74–85, Sept. 2009.

[33] H. Li and Z. Han, "Catching attackers for collaborative spectrum sensing in cognitive radio systems: An abnormality detection approach," in *New Frontiers in Dynamic Spectrum, IEEE Symposium on*, pp. 1–12, April 2010.

[34] J. Wei and X. Zhang, "Two-tier optimal-cooperation based secure distributed spectrum sensing for wireless cognitive radio networks," in *INFOCOM IEEE Conference on Computer Communications Workshops*, pp. 1–6, March 2010.

[35] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of byzantine attack in large wireless sensor networks," in *Military Communications Conference, MILCOM IEEE*, pp. 1–4, Oct 2006.

[36] A. Rawat, P. Anand, H. Chen, and P. Varshney, "Countering byzantine attacks in cognitive radio networks," in *Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference on*, pp. 3098–3101, March 2010.

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

[38] "CISCO location tracking approaches." `http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/wifich2.html#wp1049544`.

[39] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*. Prentice Hall PTR, 1st ed., 2001.

[40] J. Hillenbrand, T. Weiss, and F. Jondral, "Calculation of detection and false alarm probabilities in spectrum pooling systems," *Communications Letters, IEEE*, vol. 9, pp. 349–351, April 2005.

[41] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom, pp. 255–265, ACM, 2000.

[42] M. Vu, N. Devroye, M. Sharif, and V. Tarokh, "Scaling laws of cognitive networks," in *Cognitive Radio Oriented Wireless Networks and Communications CrownCom. 2nd International Conference on*, pp. 2–8, Aug 2007.

[43] S.-W. Jeon, N. Devroye, M. Vu, S.-Y. Chung, and V. Tarokh, "Cognitive networks achieve throughput scaling of a homogeneous network," *Information Theory, IEEE Transactions on*, vol. 57, pp. 5103–5115, Aug 2011.

[44] M. Vu, N. Devroye, and V. Tarokh, "On the primary exclusive region of cognitive networks," *Wireless Communications, IEEE Transactions on*, vol. 8, pp. 3380–3385, July 2009.

[45] in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, vol. 4724 of *Lecture Notes in Computer Science*, 2007.

[46] A. Josang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.

[47] A. Josang and J. Haller, "Dirichlet reputation systems," in *Availability, Reliability and Security, (ARES) The Second International Conference on*, pp. 112–119, April 2007.

[48] A. Josang, "Trust-based decision making for electronic transactions," in *Proceedings of the Fourth Nordic Workshop on Secure Computer Systems (NORDSEC)*, 1999.

[49] A. Rawat, P. Anand, H. Chen, and P. Varshney, "Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks," *Signal Processing, IEEE Transactions on*, vol. 59, pp. 774–786, Feb 2011.

[50] Y. Sun, Z. Han, W. Yu, and K. Liu, "A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks," in *INFOCOM 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–13, April 2006.

[51] K. Fang and Y. Zhang, "Generalized multivariate analysis,," Nov 1990.

[52] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secur. Comput.*, vol. 1, pp. 11–33, Jan. 2004.

[53] D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems (3rd Ed.): Design and Evaluation*. Natick, MA, USA: A. K. Peters, Ltd., 1998.

[54] K. Kwiat, A. Taylor, W. Zwicker, D. Hill, S. Wetzonis, and S. Ren, "Analysis of binary voting algorithms for use in fault-tolerant and secure computing," in *Computer Engineering and Systems (ICCES), International Conference on*, pp. 269–273, Nov 2010.

[55] D. Lindley, "Making decisions,," August 1991.

[56] Z. Ren, G. Wang, Q. Chen, and H. Li, "Modelling and simulation of rayleigh fading, path loss, and shadowing fading for wireless mobile networks," *Simulation Modelling Practice and Theory*, vol. 19, no. 2, pp. 626 – 637, 2011.

[57] Y. Cai, L. Cui, K. Pelechrinis, P. Krishnamurthy, M. Weiss, and Y. Mo, "Decoupling trust and wireless channel induced effects on collaborative sensing attacks," in *Dynamic Spectrum Access Networks (DySPAN), IEEE International Symposium on*, pp. 224–235, April 2014.