2015

Rogue Signal Threat on Trust-based Cooperative Spectrum Sensing in Cognitive Radio Networks

David S. Jackson
Virginia Commonwealth University

ROGUE SIGNAL THREAT ON TRUST-BASED COOPERATIVE SPECTRUM SENSING IN COGNITIVE RADIO NETWORKS

A dissertation proposal submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

David Scott Jackson,
Ph.D. in Computer Science, VCU 2015
M.S. in Computer Science, VCU 2013
B.S. in Computer Science, VCU 2011

Meng Yu, Ph.D., Associate Professor, Department of Computer Science
Wanyu Zhang, Ph.D., Assistant Professor, Department of Computer Science

Virginia Commonwealth University
Richmond, Virginia
June 2015

ACKNOWLEDGEMENT

I would like to thank Dr. Yu, Dr. Zang, and my fellow colleagues from the security lab for their support and guidance throughout my work. I would also like to thank the VCU School of Engineering for having me as a student since 2006, a whole 9 years of my life well-spent. Last but not least, I want to thank my parents for taking care of me through college and for also believing in me. A few years back, I remember telling my dentist that I was nearing graduation of my Ph.D. studies in Computer Science, and his response was "That's good! Because soon you will be a contributing member of society." That meant a lot to me, since I've been a full-time student for almost a decade in the hopes of one day putting my skills to good use. Although it may not seem like much, all the encouragement I received from friends and family really added up (dentist included). I believe it was their support that kept me going strong during graduate school. I can't thank you guys enough!

Contents

List of Figures

List of Tables

Nomenclature

H_0	Null Hypothesis - Primary Signal Absent
H_1	Alternative Hypothesis - Primary Signal Present
CR	Cognitive Radio
CRN	Cognitive Radio Networks
CSS	Cooperative Spectrum Sensing
DARPA	Defense Advanced Projects Research Agency
DSA	Dynamic Spectrum Access
FC	Fusion Center
FCC	Federal Communications Commission
IDS	Intrusion Detection System
IoT	Internet-of-Things
M2M	Machine-to-Machine
NTIA	National Telecommunications and Information Agency
PKI	Public Key Infrastructure
PU	Primary User
PUE	Primary User Emulation
QoS	Quality-of-Service
RSF	Rogue Signal Framing

RSS	Received Signal Strength
SSDF	Spectrum Sensing Data Falsification
SU	Secondary User
TVBD	TV Bands Device
WNAN	Wireless Network After Next
WRAN	Wireless Regional Area Network

Abstract

ROGUE SIGNAL THREAT ON TRUST-BASED COOPERATIVE SPECTRUM SENSING IN COGNITIVE RADIO NETWORKS

By David Scott Jackson, Ph.D.

A dissertation proposal submitted in partial fulfillment of the requirements for the degree of Ph.D. at Virginia Commonwealth University.

Virginia Commonwealth University, 2015.

Major Director: Meng Yu, Ph.D.

Associate Professor, Department of Computer Science

Cognitive Radio Networks (CRNs) are a next generation network that is expected to solve the wireless spectrum shortage problem, which is the shrinking of available wireless spectrum resources needed to facilitate future wireless applications. The first CRN standard, the IEEE 802.22, addresses this particular problem by allowing CRNs to share geographically unused TV spectrum to mitigate the spectrum shortage. Equipped with reasoning and learning engines, cognitive radios operate autonomously to locate unused channels to maximize its own bandwidth and Quality-of-Service (QoS). However, their increased capabilities over traditional radios introduce a new dimension of security threats.

In an NSF 2009 workshop, the FCC raised the question, What authentication mechanisms are needed to support cooperative cognitive radio networks? Are reputation-based schemes useful supplements to conventional Public Key Infrastructure (PKI) authentication protocols? Reputation-based schemes in cognitive radio networks are a popular technique for performing robust and accurate spectrum sensing without any inter-communication with licensed networks, but the question remains on how effective they are at satisfying the FCC security requirements.

Our work demonstrates that trust-based Cooperative Spectrum Sensing (CSS) protocols are vulnerable to rogue signals, which creates the illusion of inside attackers and raises the concern that such schemes are overly sensitive Intrusion Detection Systems (IDS). The erosion of the sensor reputations in trust-based CSS protocols makes CRNs vulnerable to future attacks. To counter this new threat, we introduce community detection and cluster analytics to detect and negate the impact of rogue signals on sensor reputations.

1 Introduction

Along with advent of the Internet-of-Things (IoT), it is envisioned that billions of machines will be connected to the Internet, pushing the current communication technologies to their limits in terms of connectivity and performance. Not too long ago, wireless technologies were generally thought of only encompassing Wi-Fi-enabled laptops, smartphones, and the emerging tablets. Now, the Internet-of-Things, also referred to as Machine-to-Machine (M2M), encompasses much more than that. This includes everything from cell phones, coffee makers, washing machines, headphones, lamps, wearable devices, and almost anything else you can think of [19]. Keeping all this in mind, two problems need to be addressed: 1) "how are we going to overcome the spectrum shortage problem to enable such interconnectivity at a large scale?" and 2) "how can we manage so many wireless devices effectively?" The Cognitive Radio technology can help mitigate interference and improve Quality-of-Service (QoS) in such environments by employing smart techniques for accessing the wireless spectrum in an opportunistic manner [48].

Cognitive Radio Networks (CRNs) can sense, detect, and monitor their surrounding radio frequency conditions including the interference and availability of a broad range of wireless channels, followed by selecting the best one for a given task. This is called Dynamic Spectrum Access (DSA) and it is a key characteristic of cognitive radios that enable Secondary Users (SUs) to operate on geographically unused channels, even when that channel frequency is licensed to Primary Users (PUs), *e.g.*, an AM/FM radio broadcast station. They have the potential to increase spectrum efficiency that leads to higher bandwidth services and reduce the burdens of centralized spectrum management by public safety communications officials [18]. The Defense Advanced Projects Research Agency (DARPA) XG and WNAN (Wireless Network After

Next) programs are investigating the potential of DSA-capable radios based on inexpensive and adaptable radio architectures that can respond dynamically to the radio's surrounding environment [18].

1.1 Spectrum Bandwidth Bottleneck

The growing demand for wireless services shows an inevitable overcrowding of the spectrum bands, in large part due to the rapid increase of wireless mobile services in recent years, as depicted in Figure 1 [24]. This example shows the spectrum demand for mobile broadband services surpassing the available spectrum as early as mid-2013, but obviously the demand can only go as high as the capacity. This example illustrates the need for innovative solutions to alter the trajectory of overcrowded spectrum bands. DSA is the proposed solution to alleviate the overcrowding of bands by allowing licensed PUs to share unused spectrum with non-licensed SUs in an opportunistic fashion [4, 13].

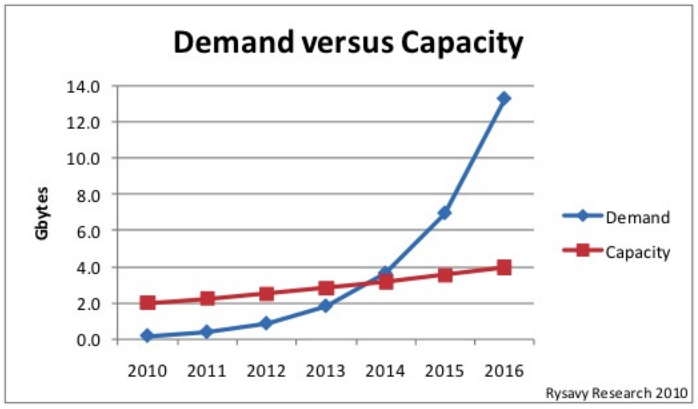


Figure 1: Spectrum Demand vs Capacity

Conventionally, the Federal Communications Commission (FCC) had statically assigned spectrum bands to PUs for exclusive use on a long term basis, precluding anyone else from access [4]. Yet, analysis of the spectrum bands clearly indicate that current FCC policies have created severely under-utilized spectrum bands, causing a bottle-

neck for new wireless applications. Figure 2 depicts these under-utilized spectrum bands across the usable radio-frequency spectrum. White space is what the FCC calls a spectrum band, i.e., a radio frequency wireless channel, that is not used by the PU.

Figure 3, from Google's spectrum database [27], illustrates that spectrum bandwidth shortage only occurs in densely populated areas, i.e. the major cities, in the United States, but there remains an abundance of white space all over the country. White space is indicated by the color green in Figure 3. The FCC is promoting a spectrum sharing paradigm, where licensed spectrum bands intended for PUs are accessible to SUs on a non-interference basis, as a way to mitigate the spectrum shortage problem [12].

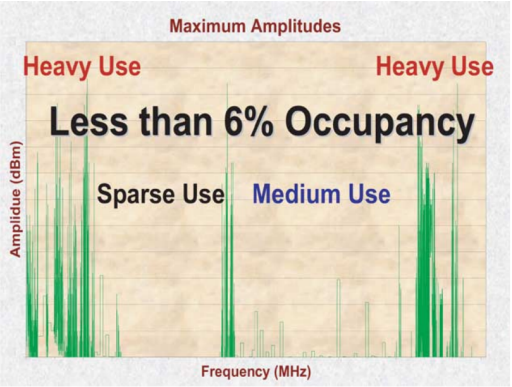


Figure 2: White space across spectrum

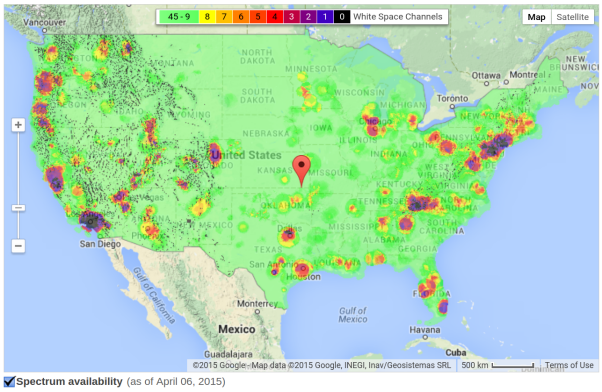


Figure 3: White space across US geography

Cooperative spectrum sensing (CSS) has been proposed as an effective approach for boosting the detection of primary signals in CR networks, so that SUs know when to yield to PUs quickly enough to avoid any interference [13, 50, 29]. In centralized CSS, the SUs submit their sensor reports to the Fusion Center (FC), which is a server for aggregating and cross-examining the network's sensor reports to make a robust analysis of the spectrum availability. The purpose of the FC is to output a global spectrum decision, based on the sensor reports, to notify SUs if a licensed spectrum band is available. CSS solves the hidden node problem where a lone SU fails to perceive the

primary signal due to shadow fading, and causes interference with nearby PUs. By working together in CSS, the lone SU can be notified of the existence of the primary signal from its neighbors [38]. Figure 4 illustrates the CSS model of wireless sensors gathering information on spectrum availability and reporting it to a fusion center for a spectrum decision. A global decision (GD) is made after each iteration of the CSS model, which is either the H_0 or H_1. The null hypothesis H_0 presumes the primary signal is absent, and the alternative hypothesis H_1 presumes the primary signal is present.

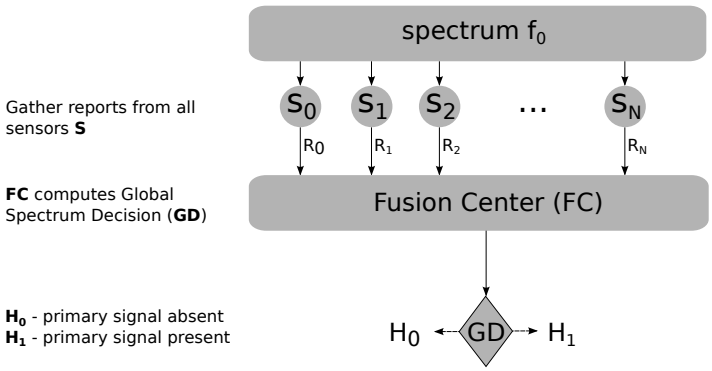


Figure 4: Diagram of Cooperative Spectrum Sensing (CSS)

However, CSS is vulnerable to attacks like the Spectrum Sensing Data Falsification (SSDF) where malicious SUs make false reports on the spectrum availability to mislead the FC. To counter SSDF, various trust models have been proposed to protect CSS from malicious SUs. These trust-based CSS protocols build reputation profiles for sensors and filter out the sensing reports from those with low reputations [12, 30, 5, 7, 25]. Thus, they can single out attackers and mitigate their influence in the shared spectrum sensing.

Depending on how rogue signals are used, they can achieve Primary User Emulation (PUE) [12], Sensory Manipulation [15], or Rogue Signal Framing (RSF) attacks [28]. Primary Use Emulation is when a secondary user masquerades as the primary user,

forcing all other secondary users to evacuate some channel, and thus invoking a Denial-of-Service attack for the secondary network [12]. The Sensory Manipulation attack occurs when spoofed (rogue) signals distort the environmental perception of a cognitive radios over time, eventually causing faulty statistics to be stored in its Knowledge Base (KB) [15]. The KB is the database of information used as input for the cognitive radio's learning and reasoning engines. In my work, I introduce the RSF attack as an exploit on trust-based CSS protocols where rogue signals gave the impression that SUs were malicious, when in fact the cognitive radio sensors were well-behaved but under the influence of unauthorized rogue signals. From this point on, I will only refer to "cognitive radio sensors" as simply "sensors" for convenience.

1.2 Cognitive Radios

Cognitive Radios (CR) are adaptive radios that are designed for improved performance and flexibility in wireless communications over the traditional radios that are built upon the more rigid Application-Specific Integrated-Circuit (ASIC) devices. Unlike their predecessors, cognitive radios can be programmed to have any of the following qualities: awareness of their operating environment and their own capabilities, autonomous operations to achieve the radio's goal, and the ability to learn and adapt from past experiences [55]. In particular, cognitive radios are well known for having autonomous frequency agility, the ability to switch channels dynamically over a broad range of radio-frequency spectrum for a more suitable connection, without the need of user interaction. In contrast, traditional radios broadcast on a single, fixed frequency channel such as the AM/FM radio stations, television networks, cell phones, and so on. In these examples, both the broadcaster and listener have to be tuned to the same frequency to receive a particular service such as music from an FM radio station. An example of a primary network consists of a TV broadcasting station (i.e.

the primary transmitter) and the corresponding subscribed viewers (i.e. the primary receivers) [46, 13].

Cognitive Radios are the devices that enable DSA due to their ability to scan spectrum bands and locate the best available channels on a non-interference basis [15]. The exact definition of cognitive radios has evolved and branched off into different meanings. The FCC defines cognitive radios as "a radio system whose parameters are based on information in the environment external to the radio system." [9] The National Telecommunications and Information Agency (NTIA) has proposed cognitive radios to be defined as "a radio or system that senses its operational electromagnetic environment and can dynamically and autonomously adjust its radio operating parameters to modify system operations, such as maximize throughput, mitigate interference, facilitate interoperability, and access secondary markets." [9] However, Joseph Mitola was the first to coin the term "Cognitive Radios" in 1999 and explained it as an intelligent agent that could search out ways to deliver services and adapt the network protocol stack to better satisfy the user's needs [37]. The key aspects associated with Mitola's vision of cognitive radios is that they are [9]:

- **Aware** of surrounding environmental conditions (e.g. the interference for some channel) and the radio's internal state such as the operational parameters for some wireless service;
- **Adapting** to its environment in real time (e.g. switching to a less noisy channel) to satisfy the requirements of some wireless service (e.g. message integrity or Quality-of-Service);
- **Reasoning** on observations to make the best known decisions, which include how to adapt to a particular scenario;
- **Learning** from previous experience to improve its reasoning capabilities; and

- **Collaborating** with other devices to make decisions based on collective observations and knowledge.

These key features require the implementation of artificial intelligence algorithms as an integral part of the CR. However, the research community remains divided on how many, and the scope of, these features a radio must possess before it is considered a CR. The first large scale standard for cognitive radios, the IEEE 802.22, is primarily focused on frequency agility that addresses the mitigation of interference to PUs [9]. Although cognitive radios are associated with frequency agility and DSA, neither of these features alone account for the main intelligent attribute that cognitive radios were initially known for.

Regardless of how cognitive radios are being interpreted, they are being pushed as the means to solve the spectrum shortage problem by utilizing much of the untapped spectrum bands as illustrated in Figure 2. The secondary network, consisting of cognitive radios, is given permission to coexist in licensed channels under two preconditions mandated by the FCC: (1) giving spectrum priority to licensed users and (2) minimizing interference to licensed users. The faster the SUs can detect the primary signal and vacate the licensed channels, the smaller the interference to the PUs, thus allowing then the secondary signals to collide less frequently with the primary signal. For this reason, the secondary network must achieve accurate spectrum sensing to know exactly when PUs occupy the channel.

Cognitive Radios are composed of several parts: the Software Defined Radio (SDR), a knowledge base, and the learning and reasoning engine. Traditional radio chips (or hardware-based) are hard-wired to communicate using one specific protocol. For example, a typical cell phone has several different chips to handle a variety of radio communications: one to contact cell phone towers, another to contact WiFi base stations, a third to receive GPS signals, and a fourth to communicate with Bluetooth devices. In

comparison, software-defined radio hardware works with raw electromagnetic signals, relying on software to implement specific applications. This makes software-defined radio devices incredibly versatile, because it has the potential, with the appropriate software, to perform the same features of all the hardware-based chips currently in our mobile devices.

Software-defined radio (SDR) is a radio communication technology that is based on software defined wireless communication protocols instead of hard-wired implementations. In other words, frequency band, air interface protocol and functionality can be upgraded with software download and update instead of a complete hardware replacement. SDR provides an efficient and secure solution to the problem of building multi-mode, multi-band and multifunctional wireless communication devices.

An SDR is capable of being re-programmed or reconfigured to operate with different waveforms and protocols through dynamic loading of new waveforms and protocols. These waveforms and protocols can contain a number of different parts, including modulation techniques, security and performance characteristics defined in software as part of the waveform itself.

Figure 5 shows a diagram of the four main components of a cognitive radio. The knowledge base is the cognitive radio's database of environmental statistics (channel noise), communication policies, and any other information that influence its actions [15]. Within the cognitive engine, there are two mechanisms for interacting with the knowledge base: the reasoning engine and the learning engine. A policy radio only has a reasoning engine, while a learning radio has both a reasoning and a learning engine. The reasoning engine is a set of logical inferencing rules, sometimes called a case-based reasoner. Learning radios typically utilize a variety of classic AI learning algorithms, including search algorithms, neural networks, and evolutionary algorithms. For example, a radio can try out different modulation types to see which works op-

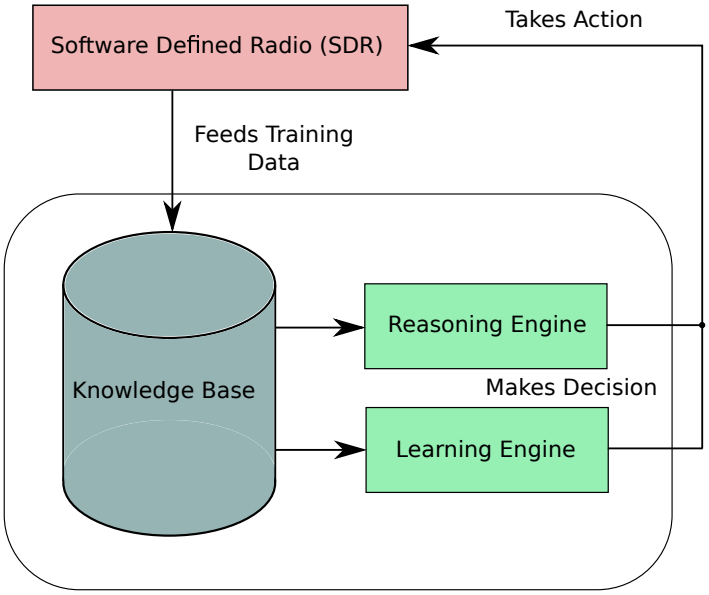


Figure 5: Diagram of a Cognitive Radio

timally in a particular RF environment [15]. Equipped with reasoning and learning engines, cognitive radios operate autonomously to locate the best unused channels to maximize its own bandwidth. However, their increased capabilities over traditional radios introduce a new dimension of security threats [21, 10].

1.3 Cooperative Spectrum Sensing

Cognitive radios utilize the DSA technology that enables autonomous optimization of radio configurations and the scanning of spectrum bands to locate the best available channels on a non-interference basis [15, 52, 53]. The cognitive radio network, consisting of SUs, is given permission to coexist in licensed channels under two preconditions mandated by the FCC: (1) giving spectrum priority to licensed users and (2) minimizing interference to licensed users. The faster the SUs can detect the primary signal and vacate the licensed channels, the smaller the interference. For this reason, the secondary network must achieve accurate spectrum sensing to know exactly when primary users occupy the channel [17].

The cornerstone of the IEEE 802.22, the first standard for cognitive radio networks, requires the SUs to yield to the PUs immediately after detecting the primary signal within a designated region [46]. The 802.22 WRAN standard is aimed at using DSA technology to allow sharing of geographically unused spectrum allocated for television broadcast services. So in the 802.22 WRAN implementation, the primary network would consist of a TV broadcasting station (primary transmitter) and the corresponding subscribed viewers (primary receivers) [46, 13]. Ideally, SUs would occupy unused TV spectrum in geographical locations where the primary network is absent, but may coexist as long as the SUs do not interfere with the subscribed viewers' reception of the primary signal. However, guaranteeing a minimal level of interference to the primary network is perhaps the biggest obstacle to the commercialization of DSA technology and a very difficult problem to solve [13]. In order to have minimal interference, cognitive radios must be able to reliably detect, in real time, the presence or absence of a primary signal from a given spectrum band. Otherwise, these cognitive radios can unknowingly transmit signals simultaneously with the primary transmitter, causing unacceptable levels of interference to nearby PUs [33].

Such unintended interference can arise from the hidden node problem. Figure 6 depicts an SU obscured from the primary transmitter due to obstacles in the environment, in what is called shadow fading. Hence, the SU continues to occupy licensed spectrum bands simultaneously with nearby PUs. Additionally, an SU may not detect the primary signal because of multipath fading. This is caused by multipath propagation, the phenomenon that results in a radio signal reaching the receiving antenna in more than one path. In other words, wireless radio signals bounce off physical obstructions, propagating into new signal copies each time, and culminate into a less audible and weaker signal at the receiver. Figure 7 depicts an SU unable to detect the primary signal due to multipath propagation.

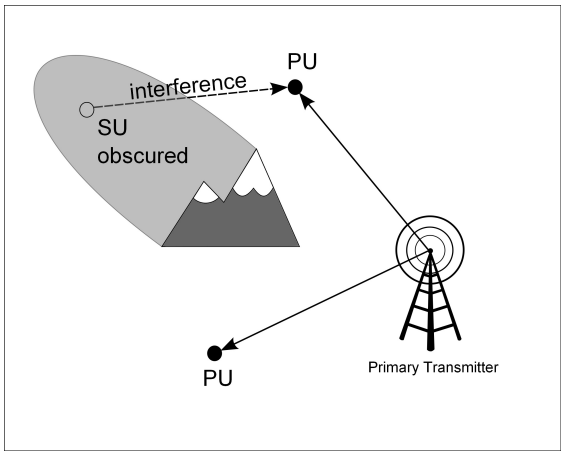


Figure 6: Causes of the hidden node problem from shadow fading

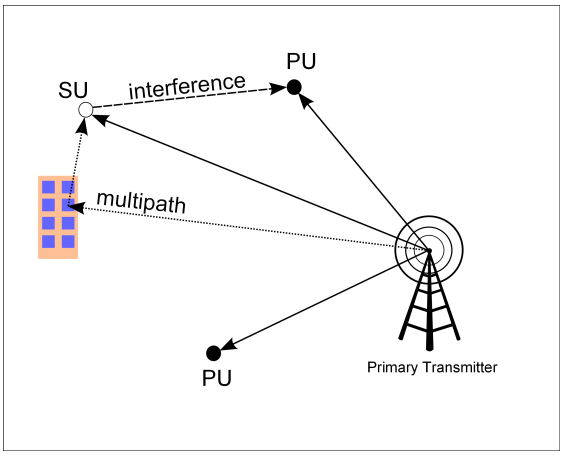


Figure 7: Causes of the hidden node problem from multipath fading

Research results from [4] indicate that shadow fading and multipath fading can be alleviated by requiring multiple SUs to cooperate with each other to conclude the spectrum availability. This collaboration of sensors, called *Cooperative Spectrum Sensing* (CSS) has been proposed as an effective approach for boosting the detection of primary signals in CR networks [36, 13, 50]. In centralized CSS, the SUs submit their sensor reports to the Fusion Center (FC), which is a server for aggregating and cross-examining the network's sensor reports for a more robust analysis of the spectrum availability. Here, the FC collects the network's sensor reports and outputs a global decision to notify SUs if they can access a licensed spectrum band [13]. In decentralized CSS, each CR operates as a local FC such that each node makes a local decision on spectrum availability based on its neighbors' data [13].

1.4 The Push for Cognitive Radio Networks

The FCC promoted the CR technology as the solution to the spectrum shortage problem under the IEEE P802.22 project that started in 2004 [26]. This project hinged on the dynamic spectrum access of CRNs to tap into geographically unused spectrum.

The key criteria to this project was the idea of a primary network, the typical licensed networks like the well-known AM/FM radio stations, and secondary networks that could harness empty spectrum in the absence of the primary networks. More specifically, they did not want to add any extra burden to the primary network, nor did the FCC want a 3rd party facilitator, but instead a self-policing secondary network [9]. The start of this project mobilized research institutions to investigate the potential for many years to come.

The FCC relies almost entirely on certification to produce trust, their process to guarantee that new devices will follow access rules through product inspection [51]. When regulators require trust, the technical response has been policy engines that are essentially easily-certifiable decision trees that guarantee a device will make certain decisions based on cognitive radio's sensory input. There are many problems with this approach, because the policy engine cannot guarantee that all contexts will produce appropriate answers. Consider all the challenges of signal detection in all environments due to shadow fading, Rayleigh fading, and interference in general. No policy engine can guarantee these nodes will be able to realize their predicament and not transmit. Hence, as of January 2011, the FCC finalized the rules for the cognitive use of TV white space in the US. Ultimately, the rules from FCC's "Second Memorandum Opinion and Order" [16] state that cognitive radios, operated by SUs, must download the spectrum occupancy table via the internet to discover unused channels.

"While we are eliminating the sensing requirement for TVBDs (TV Bands Devices), we are encouraging continued development of this capability because we believe it holds promise to further improvements in spectrum efficiency in the TV spectrum in the future and will be a vital tool for providing opportunistic access to other spectrum bands."
- FCC [16]

That means there are no longer any spectrum sensing requirements, including Cooperative Spectrum Sensing (CSS), when detecting available channels within the TV broadcast bands. However, FCC encourages continued development of spectrum sensing as stated below. Again, FCC stresses continued research in spectrum sensing applications in the following post:

"Second, I hope that equipment developers and device manufacturers will continue their work on sensing technologies and take advantage of the flexible approach outlined in the item. I appreciate the well-articulated concern that requiring both sensing and database consultation could have a chilling effect on the initial deployment of white space devices. However, I am hopeful that the widespread commercial deployment of sensing technologies will play a critical role in increasing access to spectrum not only in the TV white spaces but in other spectrum that from time-to-time or in certain locations lies fallow. Sensing technologies have shown great promise in other contexts, including Department of Defense research, and I look forward to finding ways to encourage and advance their deployment for commercial purposes." - FCC [16]

Note that the final rules encourage further research in cognitive radio sensing techniques, since this may be useful for other spectrum bands and different applications. Interestingly, the final rules discard the idea of cognitive radios: is it cognitive to download from a database a list of free channels? Although the IEEE 802.22 protocol no longer uses spectrum sensing, there are other applications that could leverage the technology. In the FCC article [18], the FCC talks about current government-funded projects aimed at new cognitive radio applications:

"As noted in our last topic, the DARPA XG and WNAN Programs are investigating much more dynamic frequency selective radios based on agile radio architectures that can respond dynamically to the radio's surrounding environment... Perhaps the key to the success and the future development of cognitive radio lies in the ability of developers and practitioners, that is, the first responder community, to establish the policy rule set by which the radios will operate..." - FCC [18]

The silver lining to the disregard of the dynamic spectrum sensing requirement, is that the FCC and DARPA still continue to promote this technology because the technology holds promise. The DARPA XG and WNAN (Wireless Network After Next) programs are investigating the potential of DSA-capable radios based on inexpensive and adaptable radio architectures that can respond dynamically to the radio's surrounding environment [18]. For example, the CR technology is believed to be able to reduce the burdens of centralized spectrum management by public safety communications officials, once the technology becomes viable [18].

The potential of the cognitive radios is being studied in many different paradigms and applications of networks, and how it can overcome many resource intensive problems. This includes cognitive mesh networks where the opportunistic spectrum access (or DSA) can alleviate the scarcity of wireless bandwidth needed to maintain the Quality-of-Service requirements [47]. Another scenario where the CR technology is being contemplated is in natural disaster areas like earthquakes and hurricanes that obliterate the devastated area's network infrastructure. The DSA from cognitive radios can temporarily provide an abundance of wireless bandwidth needed to facilitate the high traffic from emergency responders [44]. Other areas where cognitive radio applications are being researched include public safety networks [22], battlefield networks [43], and leased networks [23].

1.5 Contributions

To counteract this new threat, we propose a new defense scheme, named the RSF Clustering Defense (RCD) module, that looks for dense clusters of sensors and examines the proximity and similarity of their reports. Based on the RCD findings, it makes a heuristic decision on whether or not the network was affected by an RSF attack via rogue signals. Thus, the RCD module can distinguish sensors under the RSF intrusion and mitigate the trust damage. In effect, our defense prevents trust models from becoming an overly sensitive IDS by minimizing the false alarms caused by rogue signals, but still *relies* on a trust model to stop SSDF attacks. We focus on mitigating the impact of the RSF attack on trust-based CSS protocols by introducing a dynamic and flexible rogue signal detection solution. The following is a list of contributions:

- Introduced the Rogue Signal Framing Intrusion, an attack on the trust model of CSS protocols
- Developed a solution, the RSF Clustering Defense (RCD), that protects sensor reputations from manipulation in trust models
- Ran simulations that demonstrated the impact of the RSF intrusion and the RCD solution
- Devised a community-detection clustering algorithm to distinguish between malicious/malfunctioning sensors and well-behaved sensors that are misguided by rogue signals
- Ran extensive simulations that demonstrated an upward of 6% to 40% improvement, depending on the scenario parameters, in detecting rogue signals

The rest of the paper is outlined as follows. Chapter 2 reviews common CRN attacks and trust-based CSS protocols. Then, we present the attack model and system

in Chapter 3, and show the details and analysis of the RSF intrusion in Chapter 4. We propose the RCD defense and evaluate it in Chapter 5. Chapter 6 investigates different clustering techniques and demonstrates the effectiveness of our parameter-free solution against different scenarios. Finally, the paper is concluded in Chapter 7.

2 Related Works

My work is mostly related to the following attacks and defenses in CRNs.

2.1 PUE and SSDF Attacks

Although CRNs are vulnerable to a variety of attacks [15], two attacks received much attention. One is the Primary User Emulation (PUE) attack [11, 8, 39], where an attacker masquerades as the primary transmitter from the vantage point of its neighbors. The other attack is the Spectrum Sensing Data Falsification (SSDF) [13, 12, 45], in which compromised users falsify the local spectrum sensor reports to obscure the existence or create the illusion of a primary signal at the FC [35]. Both of these attacks attempt to deceive the FC on the availability of spectrum resources, causing networks to behave in unintended ways. In contrast, the RSF intrusion disrupts the trust between the FC and sensors, which makes the spectrum sensing less stable. Figure 8 illustrates the SSDF attack, where the grinning devil represents a malicious SU, the envelopes represent the sensor reports, and because of the falsified sensor report, the FC makes an incorrect judgement on the spectrum availability.

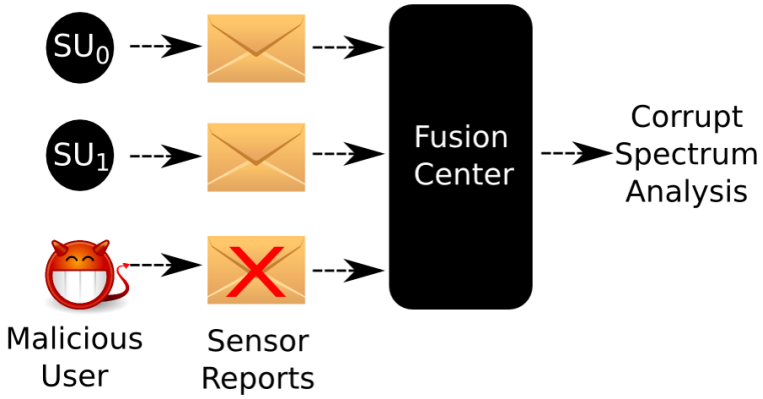


Figure 8: Spectrum Sensing Data Falsification

2.2 Trust-based CSS Protocols

To defeat SSDF attacks, various trust models have been proposed to protect CSS from malicious SUs. These trust-based CSS protocols build reputation profiles for sensors and filter out the sensing reports from those with low reputations [12, 30, 5, 7, 25]. Thus, they can single out attackers and mitigate their influence in the shared spectrum sensing. Figure 9 exemplifies the structure of the typical trust-based CSS protocol, including the trust model that filters out falsified sensor reports through cross examining the observations.

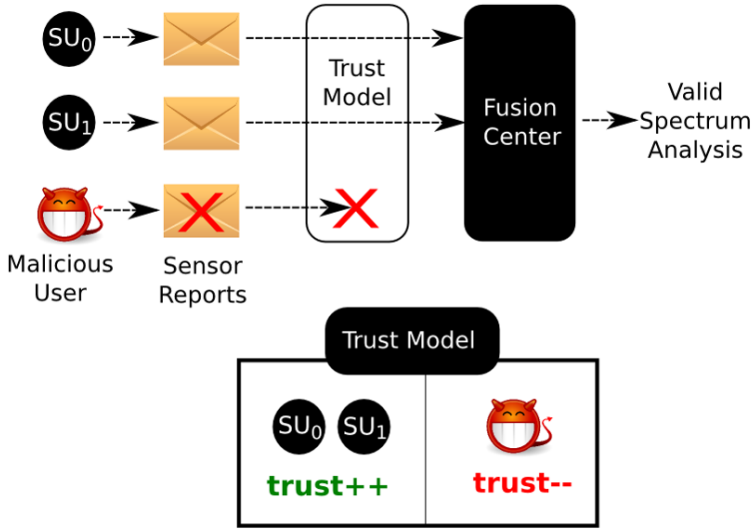


Figure 9: Trust-based CSS protocol protects CRN against SSDF attack

Chen *et al.* [12] presented a sequential probability ratio test (SPRT) that scales the contribution of sensors by their reputation in order to mitigate the impact of SSDF attacks. Their model incorporates sampling votes on the detection or absence of the primary signal, and weighing each vote according to the sensor's reputation. For every vote identical to the global decision, the sensor's reputation is incremented, such that their vote carries more weight in future decisions made at the fusion center. Kaligineedi *et al.* [30] presented a pre-filtering average combination scheme. The scheme's filters are

responsible for (1) filtering extreme outlier sensor reports and (2) ignoring sensors that have continuously deviated from the majority over a length of time. Arshad *et al.* [5] presented a beta reputation system model for hard-decision CSS protocols. Similar to [12], the sensors are rewarded for agreeing with the global spectrum decision, but otherwise penalized. In [7, 3, 54], the authors developed a trust-based CSS protocol that penalized sensors if their reports deviated too far from the expected Received Signal Strength (RSS) values determined by common RSS models. The similarity of these approaches are to build reputation profiles for spectrum sensors in order to filter out sensing reports from untrustworthy sensors. However, my work shows that the reputations can be manipulated and, as a consequence, well-behaved sensors are framed and removed from the shared spectrum sensing.

2.3 Received-Signal-Strength Anomaly Detection

Apart from reputation profiles, there are solutions that rely on RSS models and statistical methods to validate the authenticity of sensor reports. Min *et al.* [34] presented an algorithm that analyzes sensor clusters and their RSS correlation, based on distance and approximated shadow fading, to pinpoint malicious sensors and reduce/remove their input from the fusion center. A big difference in my work and theirs is that they rely (and assume) apriori knowledge of the environment's shadow fading to accurately predict the expected RSS value for a cluster of sensors. Secondly, they have no reputation model to go along with anomaly detection, so their solution discards the sensor reports in single intervals instead of penalizing the sensors for an extended duration. In [35, 32], the authors developed solutions using RSS estimation models and Support Vector Machines (SVMs), a machine learning technique, to classify sensors as either anomaly or normal. Unlike the various aforementioned solutions, I developed my own defense based on cluster analysis and community detection to safeguard sensor

reputations from manipulation, instead of only focusing on the integrity of the CCS.

What makes my solution unique is that the proposed defense protects the integrity of trust models, i.e. sensor reputations, from rogue signal manipulation. Previous literature used trust models to stop malicious SUs (and their sensors) from deceiving the CSS, but did not consider the trust models themselves to be the target of attacks. Trust models were considered reliable solutions against SSDF attacks and malfunctioning sensors, but to my knowledge, none of the papers discussed how to manipulate and disrupt trust models. I realized the vulnerability of trust models due to their coarse threshold of penalizing inaccurate sensor reports, i.e. a sensor is deemed untrustworthy if it does not behave in a predetermined way. However, if an attacker knows how the sensors should behave, then they can leverage rogue signals to disrupt typical sensor behavior and thus destroy their reputations. To protect sensor reputations, I explored techniques from social network analytics, such as cluster analysis and community detection, as opposed to relying on RSS models or shadow fading estimations to predict the correct sensor report.

Figure 10 illustrates how rogue signals can masquerade as an SSDF attack, i.e., mimicking a malicious sensor when in fact the sensors are well-behaved and functioning properly. The root of the problem lies in the trust-based CSS protocol's inability to distinguish the source of a bad sensor report, which could be due to a malicious SU, a malfunctioning sensor, interference due to shadow fading, or a purposely injected rogue signal. Protocols that punish with a broad stroke any sensor who reports differently gives attackers an exploit to turn the reputation schemes against their own users.

2.4 Motivation for Distinguishing Between RSF and SSDF

In an NSF 2009 workshop, the FCC had raised the question, "What authentication mechanisms are needed to support cooperative cognitive radio networks? Are

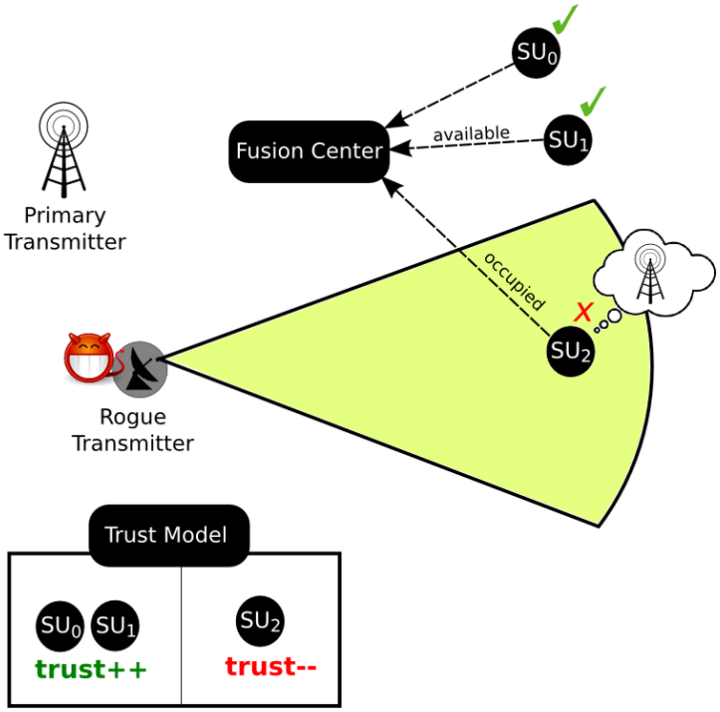


Figure 10: Trust-based CSS protocol exploited by RSF attack

reputation-based schemes useful supplements to conventional Public Key Infrastructure (PKI) authentication protocols?" [47] Reputation-based schemes in CSS (a.k.a. trust-based CSS protocols) are a popular technique for performing robust and accurate spectrum sensing without any inter-communication with the primary network, but the question remains on how effective they are at satisfying the FCC security requirements. My work takes a closer look at the robustness of trust-based CSS protocols.

In secondary networks, it is very hard to conclude the root cause of bad sensor reports, which can vary from (1) malfunctioning sensors, (2) the hidden node problem, (3) SSDF attacks (i.e. malicious secondary users), and (4) rogue signals. Yet, the trust-based CSS protocols treat all inaccurate sensors the same way, in that they penalize secondary users and diminish sensor reputation all the same. An important question I wanted to investigate was, *"Should the trust-based CSS protocols treat all inaccurate sensor reports the same way, regardless of the root cause? Or does it cause more harm*

than good to the system in certain scenarios."

To test my hypothesis, I simulated multiple directional rogue signals against targeted clusters in a cognitive radio network. The simulation illustrated the impact of rogue signals negatively affecting sensor reputations which, in severe cases, shows roughly 40% of sensors penalized and eventually ignored in the shared spectrum sensing process. In other words, nearly half of the sensors were removed without any fault of their own, e.g. the sensors were not malfunctioning nor behaving maliciously but were still penalized. That means an outsider has the potential to trick the reputation scheme in order to filter out nearly half of the sensors, thus diminishing the performance of the network's shared spectrum sensing. Trust-based CSS protocols have proven effective against malicious secondary users who report falsified sensing reports, but they did not consider the impact of rogue signals. Hence, based on the outcome of my simulations, I consider trust models as overly sensitive Intrusion Detection Systems (IDS) for penalizing sensors without taking into account the root cause of abnormal sensor reports.

Not being able to determine the origin of inaccurate sensor reports opens the possibility for attackers to use RSF as a stepping stone attack against trust-based CSS protocols. Chen et. al [12] models attacks against CSS protocols as a Byzantine Fault Tolerance system, in that the CSS protocol can continue functioning as intended as long as there are not too many Byzantine failures, which in this case are generally hidden, malicious, or malfunctioning sensors. In contrast, my work demonstrates that the RSF attack lowers the Byzantine Fault Tolerance of trust-based CSS protocols, due to having less secondary users participate in the shared spectrum sensing, thus making the system less robust against Byzantine Failures.

Clancy *et al.* [15] warns of a similar threat of rogue signals, but in a different context. They claim that rogue signals can cause faulty statistics, collected from the physical

layer (e.g. RSS, channel availability, etc.), and stored in the knowledge base. The cognitive radio's behavior is determined by the learning and reasoning engines which, in turn, depends on the knowledge base of spectrum observations across many channels overtime. Hence, the cognitive radio may not behave as intended, or in fact cause harm, when the knowledge base contains faulty statistics that inhibits good decision making. Both my work and theirs [15] express the importance of being able to defend against rogue signals. The difference, however, is my work protects the sensor reputations in trust-based CSS protocols whereas their idea is related towards protecting the integrity of the knowledge base.

3 Attack Model

In this chapter, I define the RSS model and the method of attack for the RSF which employs *directional antennas*. The attacker manipulates sensor reputations by transmitting rogue signals to targeted sensors, thus causing conflicting sensor reports in the network. To ensure that reports do conflict, directional antennas are used to avoid targeting the entire network.

3.1 System Model

Figure 11 illustrates the system model of trust-based CSS protocols and the different targets of PUE and RSF intrusions. In it, f_0 represents some wireless spectrum frequency, S_i a set of sensors, and R_i the corresponding set of sensor reports. The system model is a stack of dependent layers, starting with the spectrum channel, the network of sensors, the trust model, and finally the FC. The accuracy of the CSS is dependent on the FC receiving reliable input from the above layers. For example, the spectrum

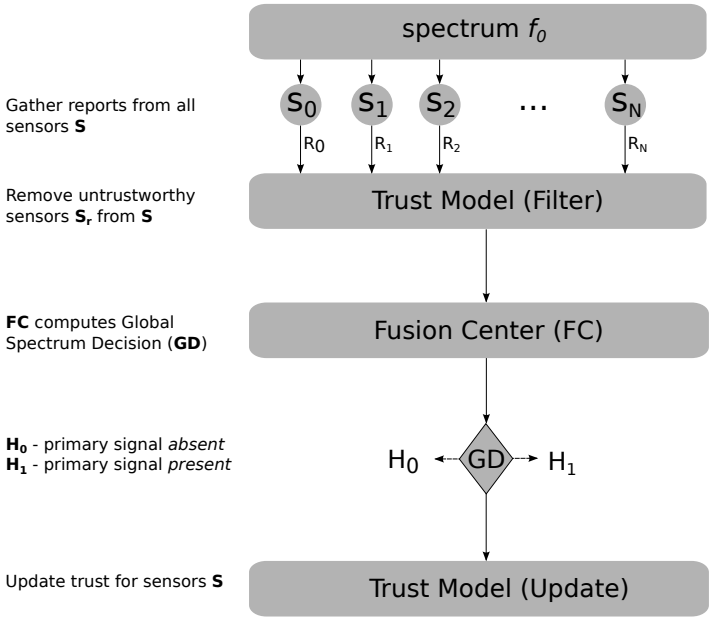


Figure 11: Trust-based CSS Protocol

channel must be clear enough for communication, the majority of sensors must not be malicious or malfunctioning, and the trust model must filter the malicious sensors to protect the FC from bad input.

Without loss of generality, I use a system as shown in Figure 12 to discuss the proposed security issues. Within the network area, the spectrum sensors are randomly distributed and the attacking antennas are positioned in the middle. The FC collects the sensor reports and cross-examines the local spectrum observations to make a global decision on channel vacancy. Spectrum sensing occurs in scheduled time intervals when all communications from the secondary network stops, called *quiet periods*, in order to listen for the primary signal [13].

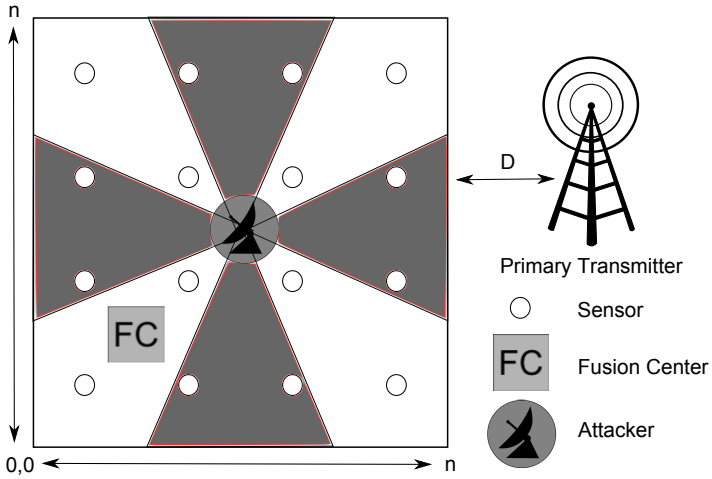


Figure 12: RSF Attack Model

Computer generated simulations were used to demonstrate my hypothesis, i.e., the vulnerabilities inherent in trust-based CSS protocols. The reason for using simulations was to create the same environment assumed in the first ever cognitive radio standard, the IEEE 802.22 WRAN standard [46], which has a very large contour region and network size in terms of CR users. Currently, there does not exist a test bed of cognitive radios that compares in size and scope of the IEEE 802.22 environment, nor was there any datasets that suited the needs of this paper from well-known data repositories like

CRAWDAD [1].

3.2 Propagation Model

Energy detection. I decided to use energy detection because it is the most widely used spectrum sensing technique for cognitive radio networks [34, 50, 42]. Secondly, energy detection is used on three trust-based CSS protocols that I borrow for my simulations, from papers [12, 30, 5].

When an attacking antenna emits signals, the RSS in decibels per milliwatt (dBm) for any given sensor s_i can be modelled according to [41]:

$$R_i = \begin{cases} \mathcal{N}(\mu_\omega, \sigma_\omega), & H_0 \\ 10\log_{10}(P_{\text{ray}}(d_{ij})) + L_s[x_i, y_i], & H_1 \end{cases} \tag{1}$$

This model gives two possible RSS values. When the antenna is not transmitting (i.e., case H_0), the RSS is simply the environmental noise, for which μ_ω is the noise power mean and σ_ω is the noise variance. On the other hand, when the antenna is emitting signals (i.e., case H_1), the RSS is determined by the attenuation of signal propagation from the attacker to the sensor plus shadow fading on position $[x_i, y_i]$. The function $10\log_{10}(\cdot)$ is used to convert milliwatts to dBm.

In the H_1 case, I use the Rayleigh fading model in milliwatts (mW), expressed as: [31, 49]

$$P_{\text{ray}}(d_{ij}) = P_{\text{FS}}(d_{ij})\sqrt{r_1^2 + r_2^2} \tag{2}$$

coupled with the Free Space propagation model [31]:

$$P_{\text{FS}}(d_{ij}) = \frac{P_t G_t G_r \lambda^2}{(4\pi d_{ij})^2} \tag{3}$$

where d_{ij} is the distance between s_i and the jth attacking antenna, λ denotes the wavelength (meters), P_t is the emission power, G_t and G_r are the antenna gains of the transmitter and receiver (respectively), and $r_1, r_2 \sim \mathcal{N}(0, 1)$ are used to simulate the stochastic nature of wireless channels [31]. Equation 3 is illustrated in Figure 14, with operational parameters of 10 dBm ($P_t = 10$) and 45° beamwidth ($G_t = 32$).

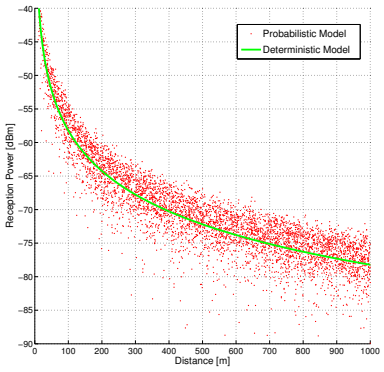


Figure 13: Propagation Model

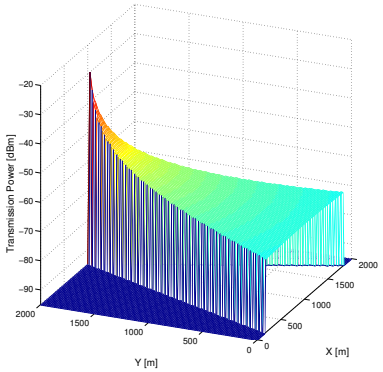


Figure 14: 3d Power Flux Density

The RSS value R_i is measured in decibels per milliwatt (dBm). However, the Rayleigh fading model (from Equation 2) is in milliwatts (mW), so I apply the unit conversion dBm $= 10 \log_{10}(mW)$ in Equation 2 under hypothesis H_1. To incorporate shadow fading into Equation 1, I used $L_s[x_i, y_i] \sim \mathcal{N}(0, \sigma_L)$ where σ_L is the shadow fading variance [20], as illustrated in Figure 15. In the propagation model, I assume that the channel bandwidth is much larger than the coherent bandwidth, so the effect of a multi-path fading is negligible, and thus removed from Equation 1 [46].

3.3 Directional Antenna Model

Rogue signals are generated by directional antennas to manipulate the sensor reputations. The antenna radiates in a smaller area surface, compressing the radiated energy, and thus raising the signal's strength. Hence, G_t in Equation 2 is substituted by the

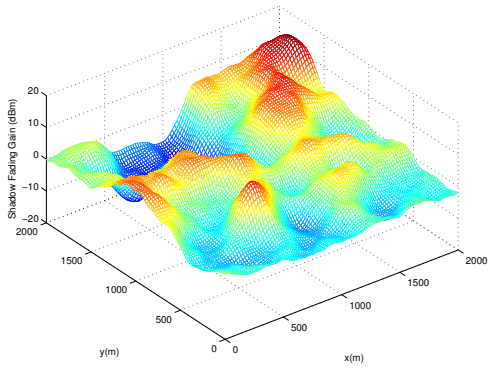


Figure 15: Auto-correlated Shadow Fading Map

directional gain according to [2]:

$$G(\theta, \phi) = (4\pi r^2)\left(\frac{4}{\pi r^2 \sin(\theta)\sin(\phi)}\right) \tag{4}$$

In Equation 4, θ and ϕ are the vertical and horizontal angles of the beam width, respectively. For simplification, I assume $\theta = \phi$. Furthermore, I assume that the rogue signals only affect the sensors inside the beams of the directional antennas. To determine which sensors are attacked, I need to calculate the angle between the attacked sensor and the directional antenna, as illustrated in Fig. 16. The angle between position $\vec{p_i}$ of the i^{th} sensor and position $\vec{p_j}$ of the j^{th} antenna is:

$$\theta_{ij} = \arccos\left(\frac{\vec{p_i} \cdot \vec{p_j}}{\|\vec{p_i}\|\|\vec{p_j}\|}\right) \tag{5}$$

where $\vec{p_i}, \vec{p_j} \in \mathbb{R}^2$. The i^{th} sensor is affected by the rogue signal if θ_{ij} falls between the lower and upper beam angles θ_l, θ_u of the j^{th} transmitter such that $\theta_l \leq \theta_{ij} \leq \theta_u$.

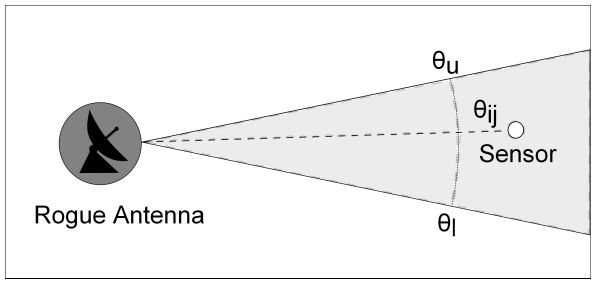


Figure 16: Capturing sensors in the radiation pattern of rogue signals.

4 Rogue Signal Framing Intrusion

In this chapter, I introduce the Rogue Signal Framing (RSF) intrusion and demonstrate its impact on the network's total trust through simulations.

In the CSS paradigm, the physical layer (i.e. the sensor) provides local signal detection. The FC collects the sensor reports and validates the signal authenticity through cross-examination of the RSS spatial diversity from the network. However, verifying the source of RF waves at the physical layer is incredibly challenging, especially for energy detectors that can only observe the RSS. Since the energy detectors only measure raw RF energy, there is no cryptographic means to identify the source [15, 38].

According to the first CRN standard, the IEEE 802.22, the secondary network must be self-reliant in minimizing interference to the primary network which requires accurate spectrum analysis [11]. In the case of SSDF attacks, trust models have been effective at removing malicious sensors from the shared spectrum sensing [12, 30, 5, 7]. However, these trust models cannot distinguish between malicious sensors and accurate sensors misled by rogue signals (as opposed to the legitimate primary signal). In other words, sensors are labeled untrustworthy when they have a consistent history of abnormal sensor reports, regardless of the cause.

Rogue signals can raise a sensor's RSS well above what is expected, especially in the absence of the primary signal. So a prolonged rogue signal on a group of sensors

can cause a sharp contrast in local spectrum observation from the others, thus appearing malicious and no different than SSDF. Consequently, the security protocol brands these sensors as untrustworthy and removes them from the shared spectrum analysis for as long as the stigma remains. As such, launching rogue signals on specific regions of the network over many quiet periods leads to the exploitation of the trust model via the RSF attack. In the context of CSS, I define the term *Rogue Signal Framing* attack as follows,

Definition: *Rogue Signal Framing* attack breaks the trust between the fusion center and a group of sensors via rogue signals to create the illusion of malicious sensors

To launch this attack, I exploit directional antennas to launch rogue signals on a regional group of sensors, and thereby causing them to report abnormally high RSS compared to the rest of the unaffected network. When sensors start reporting differently, the FC interprets the situation as an SSDF attack, when in fact the sensors reported honestly. In essence, I can use rogue signals to emulate false SSDF attacks to harm innocent sensors, and mitigate their cooperation in shared spectrum sensing.

4.1 Motivation for Directional Antennas

In a CRN with energy detectors, the RSF attacker must limit the rogue antenna's coverage in order to avoid a successful PUE. Directional antennas make it possible to isolate its radiation pattern to a targeted group of sensors (with the rest of the network unaffected), thus convincing the FC that the defecting sensors are malicious. On the other hand, isotropic antennas emit RF waves in all directions and maximize the antenna's coverage. This leaves a massive RF finger print in a network of energy detectors. Chen et. al. [11] proposed an RSS-based location verification scheme to

detect and pinpoint PUE attacks enforced by a dense network of sensors. However, this scheme was not tested or tailored for pinpointing directional antennas.

Directional antennas are difficult to detect, and even harder to pinpoint, because of their ability to emit rogue signals with narrow and asymmetrical radiation patterns. Any changes made to the beam-direction and beamwidth of a directional antenna can drastically change the network's RSS spatial diversity. These observations are supported by work from Bauer et. al. [6]. In their experiments, they demonstrated that directional antennas can disrupt localization algorithms on IEEE 802.11 WLANs that resulted in very high errors.

4.2 Trust Damage

The main goal of the RSF attack is to compromise the trust between the FC and network sensors. To quantify the trust damage (as a percentage), I use the following equation to measure the network's trust score $T_\Sigma[q]$ on quiet period q with:

$$T_\Sigma[q] = \left(\frac{1}{\sum_{s_i \in S} t_i[0]} \right) \sum_{s_i \in S} t_i[q] \tag{6}$$

where $t_i[q]$ is the trust score of sensor $s_i \in S$. In each trust-based CSS protocol, the trust score is represented differently. In order to compare the trust damage between each protocol, I normalized the trust score t_i such that $t_i[q] \in [0, 1]$ in the equation.

In each quite period, a group of sensors may lose their trust due to the RSF intrusion, so $T_\Sigma[q]$ changes from one quiet period to the next. As the time passes on, sensors exposed to RSF suffer an increasing amount of trust damage, so I expect $T_\Sigma[q]$ will decrease as the number of quiet periods q increases.

Table 1: Simulation Parameters

Parameter	Value	Description
N_s	400	Number of sensors
N_r	4	Number of rogue antennas
γ_θ	-92 dBm	Sensor sensitivity
f	615 MHz	Channel frequency
μ_ω	95.2 dBm	Noise power mean
σ_ω	0.3 dB	Noise power std
d_θ	150 m	distance threshold
σ_L	4.5 dB	Shadow fading variance
$N_x \times N_y$	$2,000$ m $\times 2,000$ m	Grid dimensions
$C_{\min}$	5	Minimum cluster size
Z_θ	0.3	Cluster threshold

4.3 Attack Evaluation

To test my proposed framing intrusion, I borrow three different trust-based CSS protocols. The first protocol F_A, by Chen *et al.* [12], utilizes the sequential probability ratio test (SPRT) and weights the probability by the sensor's reputation to mitigate the impact of SSDF attacks. The second protocol F_B, by Kaligineedi *et al.* [30], utilizes a pre-filtering average combination scheme. These filters are responsible for (1) filtering extreme outlier sensor reports and (2) ignoring sensors with high trust penalties. The third protocol F_C, by Arshad *et al.* [5], utilizes a beta reputation system model for hard-decision CSS protocols. Like F_A, the sensors are rewarded for agreeing with the global spectrum decision, but otherwise penalized. These protocols were denoted with the letter F to represent the fusion algorithm with these protocols, which is the the systematic process of collecting sensor reports and "fusing" them into a pot to make statistical observations that leads to a conclusion.

These CSS protocols were chosen because they each had different methods of evaluating trust (fusion algorithm), yet shared similar properties in which to compare them by, such as they are all centralized CSS protocols and each sensor is assigned a trust

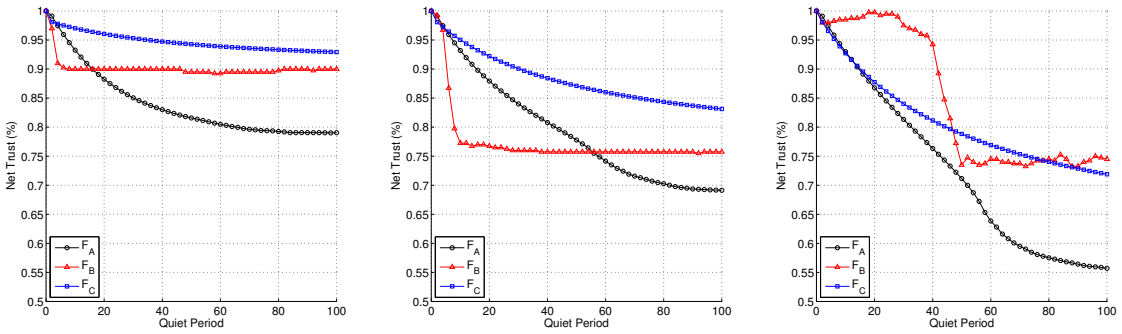
Table 2: Comparison of Selected Fusion Algorithms (CSS Protocols)

Protocol	**Fusion Algorithm**	**Sampling**	**RSS Threshold**
F_A	Sequential Probability Ratio Test (SPRT)	true	static
F_B	Average Combining, Outlier Filter	false	dynamic
F_C	Beta distribution	false	static

(or reputation) score. Table 2 shows some of the main differences between the three chosen protocols. Because protocols F_A, F_B, and F_C are centralized, I can look at the big picture that allows me to compute the network's overall statistics and apply community detection via clustering techniques. Some of the differences include: a) looking at samples vs. population of sensor reports and b) assigning an RSS threshold statically vs. dynamically for determining the FC's decision of H_0 and H_1.

I make the following assumptions on the simulation's environment according to an IEEE 802.22 WRAN environment that encompasses UHF/VHF TV bands between 54 MHz and 862 MHz [46]. In my simulation, 400 sensors are located inside a 2000×2000 grid. I assume the incumbent broadcasting station operates at the UHF frequency of 615 MHz. Like Figure 12, there are four rogue directional antennas facing the cardinal directions and positioned on the map's center. Protocols F_A, F_B, and F_C are tested on RSF attack scenarios, labeled as RSF-15, RSF-30, and RSF-45 which corresponds to the scenario's antenna beamwidths of 15°, 30°, and 45°, respectively.

Figure 17 shows the network's total trust $T_\Sigma[q]$ over 100 quiet periods for each scenario. Depending on the protocol and different evaluation environment, the RSF intrusion removed nearly 15% to 45% of the network's total trust which correlates to the percentage of sensors removed from the shared spectrum sensing. As expected, $T_\Sigma[q]$ initially decreases and plateaus over time. It plateaus when the misled sensors eventually have no more trust to lose.



(a) RSF-15 (b) RSF-30 (c) RSF-45

Figure 17: Displays the network's total trust (from Eq. 6) over 100 quiet periods for protocols F_A, F_B, and F_C. Like Figure 12, there are four rogue directional antennas facing the cardinal directions and positioned on the map's center. The beamwidth of each rogue antenna is 15°, 30°, and 45° for scenarios RSF-15, RSF-30, and RSF-45, respectively.

In Figure 17, the change in the network's total trust $\Delta T_\Sigma[q]$ per quiet period is different for protocols F_A, F_B, and F_C, because a sensor's trust score is adjusted differently for each protocol. Hence, these protocols behave differently against rogue signals, but the overall trend is a net loss of total trust $T_\Sigma[q]$ as the quiet period q increases over time. The protocol differences can be summarized briefly as follows:

- **Protocol** F_A: sensor trust is increased when the local spectrum decision agrees with the FC's global spectrum decision and penalized otherwise; only applies to a random sample of sensors with varying sizes

- **Protocol** F_B: the rate and scope of trust damage depends on the environment's RSS variance; the protocol's penalty threshold scales with the environment's noise variance

- **Protocol** F_C: sensor trust is increased when the local spectrum decision agrees with the FC's global spectrum decision and penalized otherwise; applies to all sensors

From Figure 17, I observe that both protocols F_A and F_C start to plateau, because the t_i of misled sensors eventually falls to 0, causing the $\Delta T_\Sigma[q]$ to become stagnant over time. However, protocol F_B differs in that it does not have local spectrum decisions to compare to FC's global spectrum decisions. Instead, it determines if a sensor is malicious when the reported RSS value exceeds a dynamic threshold that correlates with network's RSS variance. As the attack coverage increases from RSF-15 to RSF-45, so does the RSS variance and the F_B's behavior towards the RSF attack.

4.4 Byzantine Fault Tolerance

The CSS paradigm can be modeled in the context of the Byzantine Fault Tolerance problem. The authors in [12] describe a Byzantine failure as either a malfunctioning sensor or an SSDF attack. In both cases, the sensors perform unreliable local spectrum sensing that could ultimately mislead the FC to a wrong spectrum decision in the form of a misdetection or false alarm. These decisions are based on the null hypothesis H_0, where the primary signal is presumed absent, and the alternative hypothesis H_1, where the primary signal is presumed present, from equation 1.

A misdetection is when the FC decides H_0 when in fact the primary signal is present, and may result in unacceptable interference to the primary users. Conversely, a false alarm is when the FC decides H_1 when the primary signal is absent, and causes a Denial-of-Service of spectrum resources for secondary users. The hypothesis tests are represented in Table 3.

Table 3: Hypothesis Test

	Primary Signal Absent (H_0)	Primary Signal Present (H_1)
H_0 is accepted	Correct Decision	**Misdetection**
H_0 is rejected	**False Alarm**	Correct Decision

The RSF's ability to damage sensor reputations does not directly influence the FC's

spectrum decision like in SSDF or PUE attacks. Instead, the RSF lowers the system's *fault tolerance*, because the FC has to rely on less sensors to infer the presence of the primary signal. Hence, the RSF weakens the reliability of shared spectrum sensing for trust-based CSS protocols in the aftermath of the intrusion.

The global spectrum decision is typically determined by a consensus on spectrum observations. However, the more sensors that report inaccurate or are ignored will diminish the chance of the fusion center outputting the correct decision. Figure 18 illustrates this notion of the Byzantine Fault Tolerance in the CSS context, where the number of accurate sensors in service increases its robustness, and vice versa. Theoretically, a CSS system with hard-decision requires 51% or more sensors to swing the fusion center's decision in favor of the majority, but the percentage shrinks when sensors are filtered for bad reputation. For example, when 60 sensors out of 100 are considered trustworthy, only 31 is needed to determine the fusion center's decision.

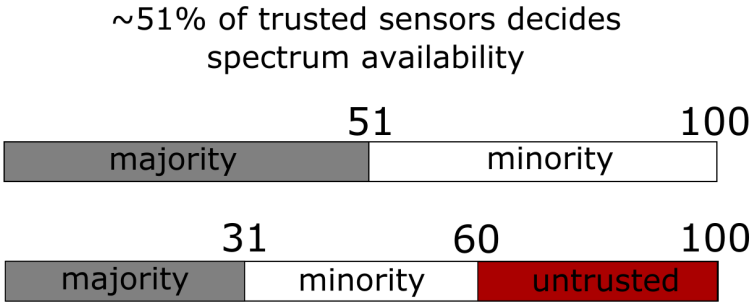


Figure 18: Byzantine Fault Tolerance applied to CSS context

I demonstrated the RSF attack via rogue signals could destroy sensor reputations earlier in this section. The next step is to measure the Byzantine Fault Tolerance of the trust-based CSS protocols after the deterioration of the sensor reputations. To evaluate the weakening of the Byzantine Fault Tolerance, I compare how many sensors need to be attacked, denoted as S_A, before the fusion center (FC) outputs an incorrect decision after four scenarios; NONE, RSF-15, RSF-30, and RSF-45. The scenario NONE is the

'no prior attack' scenario, and is used as the Byzantine Fault Tolerance in which to compare the RSF attack damages to the CSS. The RSF scenarios were conducted over 100 quiet periods. The number of attacked sensors, S_A, was incremented until the FC outputted a wrong decision. This adequately portrays either a PUE or SSDF attack, since both attacks require misleading a certain amount of sensors before becoming successful.

Table 4: Shows the number of attacked sensors S_A and safe sensors $S - S_A$

	S_A	$S - S_A$
NONE	0	400
RSF-15	40	360
RSF-30	99	301
RSF-45	169	231

Table 4 shows the number of attacked sensors S_A and, essentially, the remaining sensors left to participate in the shared spectrum sensing, $S - S_A$. Conceptually, S_A represents the number of sensors removed from contributing to the shared spectrum sensing, so the trust-based CSS protocols must rely on a smaller set of sensors ($S - S_A$).

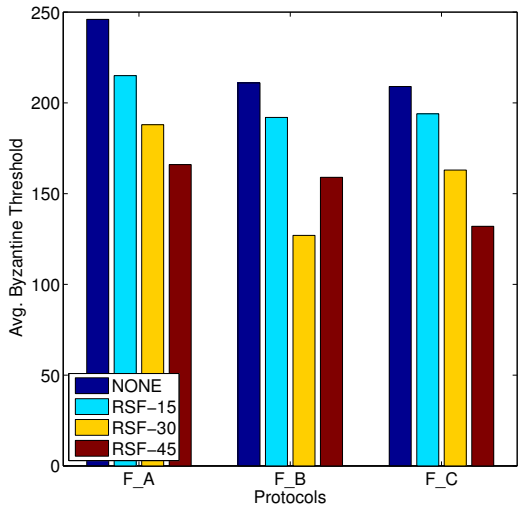


Figure 19: The Byzantine Fault Tolerance threshold of protocols F_A, F_B, and F_C

Figure 19 displays the number of attacked sensors needed (for a given protocol) to mislead the FC's spectrum occupancy decision, denoted on the y-axis title as "Byzantine Threshold". It shows that as S_A increases, the Byzantine Fault Tolerance threshold decreases, giving way to a more vulnerable CRN. Notably, the sensor reputations are exploited to weaken the system's overall robustness when rogue signals are injected. The problem is these trust-based CSS protocols cannot differentiate between SSDF attacks, which it was intended for, and RSF attacks. As pointed out in my Type-1 vs. Type-2 framing subsection, roughly 51% or more trusted sensors are needed to sway the FC's decision. Hence, for RSF-15 attack scenarios which destroys the reputations of 10% (i.e., 40) sensors, I can expect at least half of the remaining trustworthy sensors (360) to cause the FC to make an incorrect decision. This half of about 180 that sways the FC's decision represents the Byzantine Fault Tolerance threshold. The RSF-45 attack scenarios account for roughly 42.5% (169) sensors reputations destroyed, which requires roughly 116 attacked sensors to mislead the FC into making a wrong decision. In the experiments, however, the Byzantine Fault Tolerance is not perfectly aligned with each protocol since they have their differences in how they compute trust score, and whether they filter OR scale a sensor's influence on the FC by its sensor reputation. Protocol F_B has an unusual outcome in scenario RSF-30 that does not follow the same pattern as the other scenarios and protocols. This is because the RSS threshold of protocol F_B (that determines if a sensor made a good or bad choice) is dynamic, in that it fluctuates based on the average RSS readings.

4.5 Two Types of Framing

To create an illusion of malicious sensors, there needs to be a separate group of well-behaved sensors to delineate good-from-bad sensor reports. Unfortunately, classifying sensors as either honest or malicious is speculative, as the FCC regulations remove any

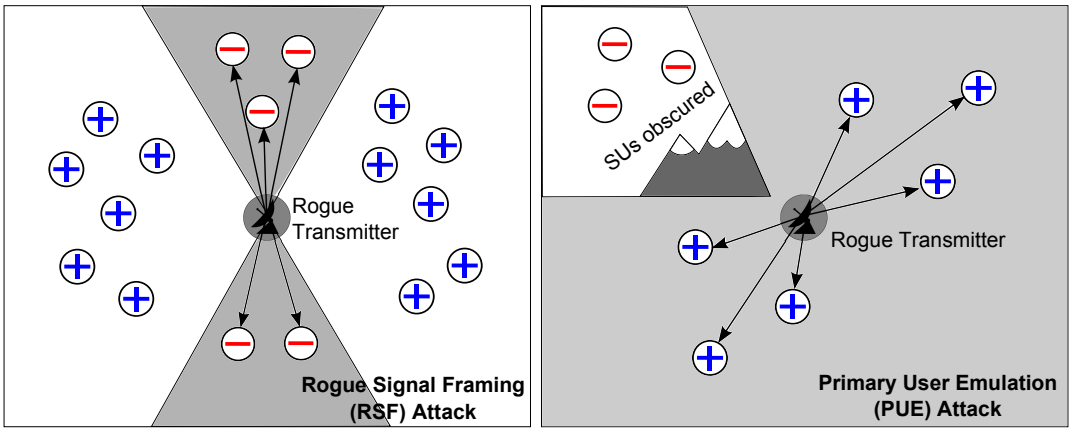
obligations of the primary network to communicate with the secondary network [15]. Hence, the secondary network is left to assume channel occupancy (i.e. the global spectrum decision) with hypotheses like H_0 and H_1. Therefore, if all sensor reputations are in good standing, such that all sensors equally participate in the shared spectrum sensing, then the global spectrum decision is typically determined by the majority of sensors.

This is especially true for hard-decision combining, which is when the FC makes a global spectrum decision based on a collection of local spectrum decisions, reported by sensors individually, in the form H_0 and H_1. Protocols F_A and F_C use hard-decision combining, with each decision weighted by sensor reputations. Alternatively, the FC can perform soft-decision combining to determine the global spectrum decision based on a collection of non-discrete sensor observations, e.g. energy detectors that report the RSS values instead of a local spectrum decision.

Soft-decision combining benefits from using more descriptive data, but also becomes more vulnerable to outliers in sensor reports, e.g. extremely high or low RSS values. Generally, CSS protocols are designed to reduce the impact of outliers or remove them entirely, but this still leaves the majority of sensor reports as a strong determinant of the global spectrum decision, just like in hard-decision combining. That is, a majority of sensors will typically decide the global decision, even if that majority is comprised of malicious sensors or affected by a wide-reaching rogue signal, as seen in the case of a PUE attack. In such a case, the FC concludes that the disagreeing minority of sensors, even if well-behaved, are presumed inaccurate.

Hence, I define two outcomes of rogue signals with regard to damaging sensor reputations, called Type-1 Framing and Type-2 Framing:

- **Type-1 Framing**: the sensors misled by the rogue signal are in the *minority* and lose trust, while the rest of the network gains trust



(a) Type-1 Framing (b) Type-2 Framing

Figure 20: The two outcomes of rogue signals in trust-based CSS protocols. The plus sign indicates an increase of reputation for some sensor, while the minus sign indicates a decrease.

- **Type-2 Framing**: the sensors misled by the rogue signal are in the *majority* and gain trust, while the rest of the network loses trust

For consistency, I will describe sensors affected by a rogue signal as *misled* sensors, and sensors that are not as *unaffected sensors*, like in Table 5.

Table 5: Attack Outcomes on Trust Models

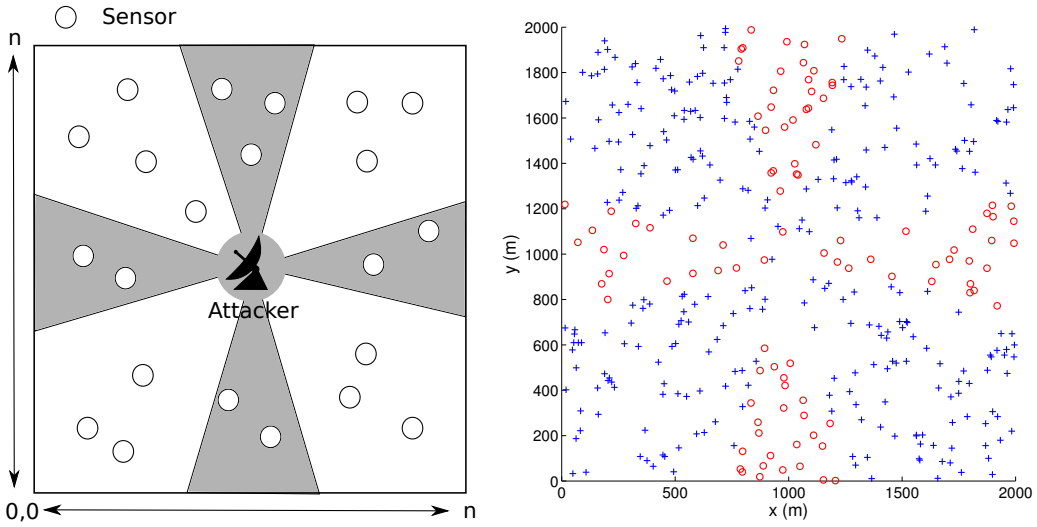
	RSF	**PUE**
Misled Sensors	*Lose* Trust	*Gain* Trust
Unaffected Sensors	*Gain* Trust	*Lose* Trust

Prior to this section, Type-1 Framing has been the designated type of trust manipulation to describe the RSF attack. Type-2 Framing, which is also a result of rogue signals, is worthy of discussion for simultaneously accomplishing a PUE attack and harming sensor reputations. Both attacks are manifested through rogue signals but can only be distinguished by the attack's outcome, such as misleading the trust model (via RSF attack) or the FC (via PUE attack). To my knowledge, the fact that a PUE attack may inadvertently affect sensor reputations has not yet been considered

in previous literature. I believe Type-2 Framing is important in that it highlights the more subtle deficiencies in trust models, like how PUE attacks can also harm sensor reputations as a side effect.

Fig 20 illustrates two cases of trust damage when the secondary network is bombarded by rogue signals: Type-1 Framing when the minority of sensors are within the attack coverage, and Type-2 Framing when the minority of sensors are outside the attack coverage. Assuming the network's trust is in a healthy state, the sensors that disagree with the global spectrum decision will be presumed malicious. In Type-2 Framing, the sensors outside the attack coverage will experience trust penalties.

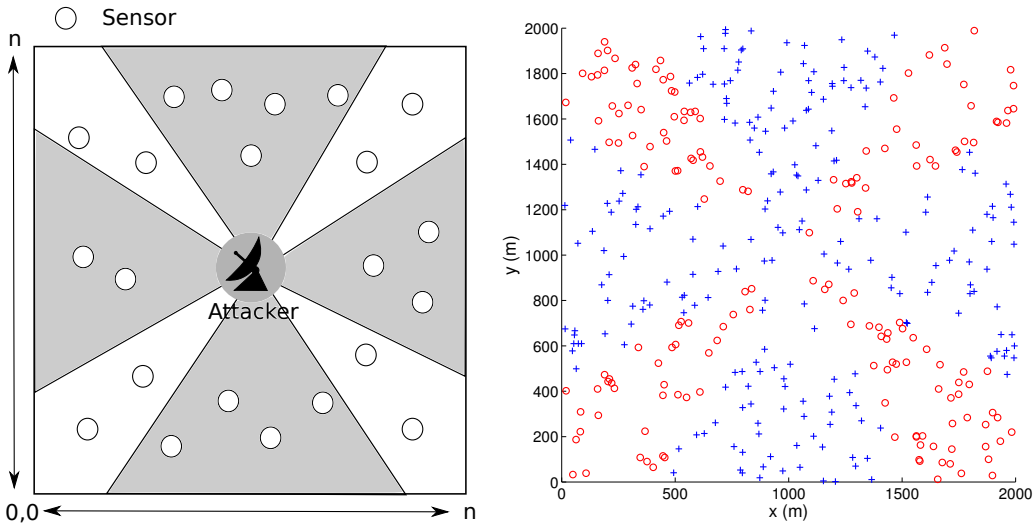
To show the two types of framing, I tested for the number of misled (attacked) sensors and PUE success rate with respect to antenna beamwidth to identify whether trust damage occurs during a PUE attack, or at least from a rogue signal with a wide attack coverage. I followed the same system parameters from Table 1. The rogue signals are launched for a duration of 100 quiet periods with a transmission power of



(a) Type-1 Diagram

(b) RSF-30 Simulated

Figure 21: Type-2 framing diagram and corresponding simulation



(a) Type-2 Diagram

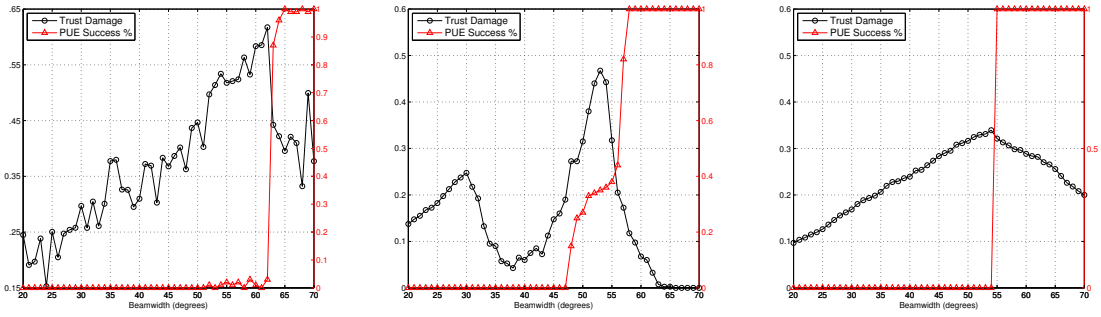
(b) RSF-60 Simulated

Figure 22: Type-1 framing diagram and corresponding simulation

10 mW for each integral beamwidth, from 20° to 70°. The recorded trust damage is based on Eq. 6 with a fixed quiet period $q = 100$.

Figures 21 and 22 depicts the simulation results of Type-1 and Type-2 Framing, respectively, on protocols F_A, F_B and F_C which shows the trust damage $T_\Sigma[100]$ (on the 100th quite period) and the PUE success rate (%) with respect to antenna beamwidth $\theta°$. Trust damage is evident in all three protocols during successful PUE attacks, i.e. when the PUE success rate is above 0. In cross examining these results, a negative correlation can be observed between the trust damage and the PUE success rate, especially upward of the 60° beamwidth mark. Hence, I use these results to reinforce the notion of Type-2 Framing as a result of rogue signals from Figure 20.

Table 6 shows the corresponding false alarms (sensors misled by rogue signals) for the beamwidth used on the four attacking directional antennas from Figure 23. The number of false alarms increases sporadically as the beamwidth increases because of the random placement of sensors.



(a) F_A (b) F_B (c) F_C

Figure 23: Trust damage over 100 quiet periods with respect to beamwidth and the corresponding PUE success rate for protocols F_A, F_B and F_C.

Table 6: Number of False Alarms for each corresponding beamwidth (degrees) from Fig. 23

Beamwidth	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°
False Alarms	56	74	100	123	143	170	190	209	229	249	283

From observing the results in Figure 23 and Table 6 as well as knowing the mechanics of the trust model algorithms, a pattern can be seen between the relationship of trust damage and false alarms. In the polar cases of 0 or N_s false alarms (where N_s is the number of sensors), the trust damage is virtually 0, since the FC cannot find any disagreements among the sensor reports.

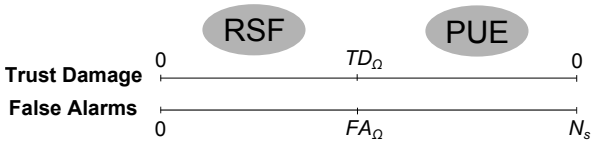


Figure 24: Modeling the Trust Damage from Figure 23

If the trust damage decreases to 0 as the number of false alarms approaches the polar ends (0 or N_s), then it can be surmised that somewhere near the middle should hold the maximum trust damage TD_Ω for a given trust model. In other words, having false alarms equal to roughly $N/2$ produces the maximum trust damage TD_Ω, because that is when the sensor network is *most divided* in local spectrum decisions. I will denote

FA_Ω as the number of false alarms that produces TD_Ω, as depicted in Figure 24.

The RSF and PUE labels over Figure 24 reflect the likely outcome of an attack from rogue signals. As the false alarms approach N_s due to rogue signals, a successful PUE attack is more likely to occur than the RSF attack. This can be observed in the PUE Success Rate in Figure 23 as the directional antennas' beamwidth broadens and the number of false alarms increases. It is important to note that regardless of the attack (RSF or PUE), trust damage occurs unless the number of false alarms is either 0 or N_s.

As seen in Table 7, the trust-based CSS protocol F_A can lose over 50% of its sensor trust (essentially removing over half its sensors) because it randomly samples sensors to make decisions, and only the sensors in the current sample are penalized if deemed inaccurate by the FC. Otherwise, protocols F_B and F_C have the same FA_Ω as a result of examining the reports of all sensors instead of sampling. The TD_Ω differs between all three protocols considering that they each use different trust-update calculations.

Table 7: Trust Model Comparison

Trust Model	FA_Ω	TD_Ω
F_A	235	63%
F_B	201	48%
F_C	201	34%

5 Clustering-based RSF Defense

This chapter introduces the RSF clustering defense (RCD) module that operates in three steps: 1) analyze the RSS diversity for any clustering behavior, 2) compute the clustering strength in order to conlude the presence of a rogue signal, and if so 3) ignore trust penalties of sensors in the attacked clusters. The defense relies on the fact that directional antennas leave isolated radiation patterns that form dense communities of sensors reporting H_1. Malicious sensors can perform SSDF attacks from the software layer without the need of rogue signals and thus operates outside the physical limitations of signal properties. In contrast, the RSF attack coverage is bound by the rogue signal's radiation pattern. Hence, I look towards a solution involving cluster analysis to exploit the rogue signal's physical characteristics, i.e., the RF "finger print" it leaves behind in a given region.

5.1 Network Classification and Clustering

The beginning of this section briefly examines the necessary network terms and concepts for better understanding the RCD algorithm and its motivation. I use graph partitioning and community detection as the basis for discovering clusters of RSF-attacked sensors. To partition the graph in a meaningful way, I assume that the nodes (e.g., sensors) have discrete characteristics such as a type or class. In my system model, the sensors are classified based on their local spectrum decision such that a given sensor s_i has a corresponding class c_i where $(c_i = -1)$ if s_i reports H_0 and $(c_i = 1)$ if s_i reports H_1. This allows for the measuring of the network's *assortative mixing*, a term defined as the pairing of nodes with the same class [40]. However, the network of sensors also needs meaningful edges for community detection. The RCD module pairs any two sensors s_i, s_j based on their class c_i, c_j and their mutual distance d_{ij} from each

other in order to observe spatial clustering.

The goal of the RCD module is to find an isolated and strongly concentrated group of sensors that report H_1. The Kronecker's delta function $\delta(\cdot)$ is a commonly used piecewise constant function in assortative mixing to specify whether or not the two nodes are of the same class [40]:

$$\delta(c_i, c_j) = \begin{cases} 0 & \text{if } c_i \neq c_j \\ 1 & \text{if } c_i = c_j \end{cases} \tag{7}$$

A basic mathematical formula for discretely measuring the assortative mixing in a network can be expressed by [40]:

$$\sum_{\text{edge}(ij)} \delta(c_i, c_j) = \frac{1}{2} \sum_{ij} A_{ij} \delta(c_i, c_j) \tag{8}$$

where c_i, c_j are the node classes and $\delta(c_i, c_j)$ is the Kronecker's delta function from Equation 7. The left side of the Equation 8 is a summation series that iterates through an edge list and increments for each pair of the same class. The right side of Equation 8 is the matrix formula which iterates through an adjacency matrix and increments the same way. The one-half fraction from the matrix formula is there to remove the double counting of pairs.

Consider Figure 25, a network with two classes of nodes such that one class is designated by black circles and the other by red squares. In such a network, a node can have a degree for each class. Each node n_i keeps track of the number of edges connected to nodes of the same class, denoted as degree k_i^{same}, as well as the number of edges connected to nodes of a different class, denoted as degree k_i^{diff}. The degree k_i^{same} can be computed by Equation 8. Similarly, the degree k_i^{diff} can be computed by

the same equation, i.e., Equation 8, with the exception of inverting the sign for the Kronecker's delta function. Figure 25 displays these two types of degrees above each node in the form of $(k^{\text{same}}, k^{diff})$ which can be used to measure the strength of the assortative mixing.

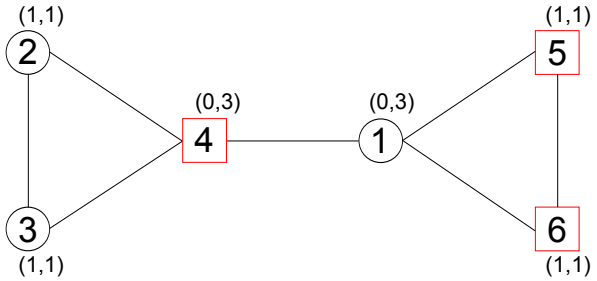


Figure 25: Example of assortative mixing.

My solution, which involves graph partitioning and community detection, is based on the principle of assortative mixing, but tailored in the context of cognitive radio networks. The RCD has three requirements for operation. First, it needs the local spectrum decision $c_i \in \{H_0, H_1\}$ for all sensors $s_i \in S$. Second, it needs two sets of sensors where $S_{H_0} = \{s_i | c_i = H_0\}$ and $S_{H_1} = \{s_i | c_i = H_1\}$. Lastly, it needs an adjacency matrix A of size $|S| \times |S|$ such that

$$A_{ij} = \begin{cases} 1 \text{ if } d_{ij} \leq d_\theta \\ 0 \text{ if } d_{ij} > d_\theta \end{cases} \tag{9}$$

where d_{ij} is the distance between sensors s_i and s_j and d_θ is the distance threshold.

The RCD module locates k disconnected clusters of sensors C_k such that $s_j \in C_k$, $A_{ij} = 1$, and $c_i = c_j$ for sensors $s_i, s_j \in C_k$. The RCD module's goal is to locate isolated communities C_k that are surrounded by sensors in S_{H_0}. To start, I measure the cluster density of sensors with the same class by counting all connected pairs (s_i, s_j) such that

$s_i \in C_k$, $s_j \in S_{H_1}$, and $A_{ij} = 1$. This is computed on all sensors in C_k with:

$$\{d_i^{H_1}\}_k = \{ \sum_{s_j \in C_k} (A_{ij}\delta(c_i, c_j)) - 1 \mid s_i \in C_k\} \tag{10}$$

where $\delta(c_i, c_j)$ is a simple Kronecker's delta function from Equation 7 that indicates a difference in a node's class c, i.e., the local spectrum decision. Next, I measure the isolation of sensor $s_i \in C_k$ from $s_j \in S_{H_0}$ by counting all connected pairs (s_i, s_j) such that $A_{ij} = 1$. This is computed on all sensors in C_k by:

$$\{d_i^{\Delta}\}_k = D(C_k) = \{ \sum_{s_j \in S_{H_0}} A_{ij}\delta'(c_i, c_j) \mid s_i \in C_k\} \tag{11}$$

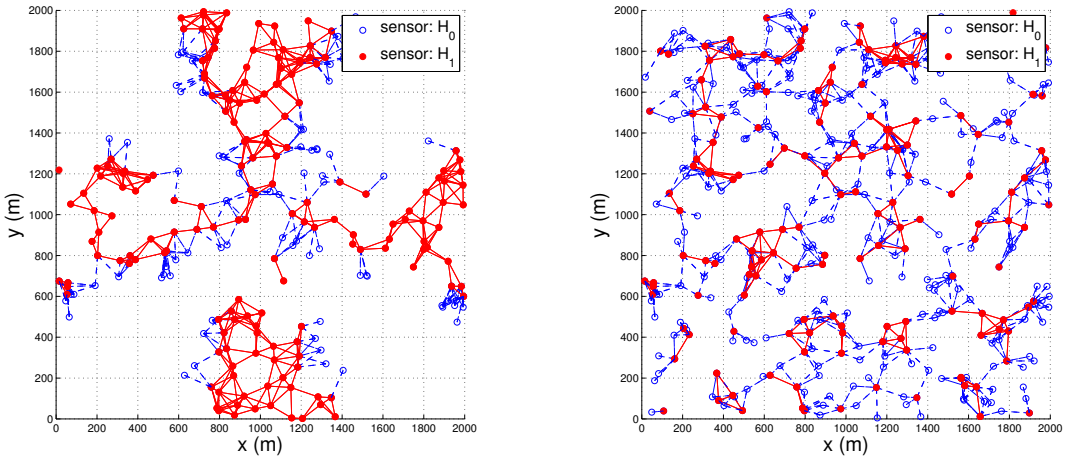
$$\delta'(c_i, c_j) = D'(C_k) = \begin{cases} 0, & \text{if } c_i = c_j \\ 1, & if c_i \neq c_j \end{cases}$$

Finally, to measure the isolated clustering strength z_k, I use the function:

$$z_k = Z(\{d_i^{H_1}\}_k, \{d_i^{\Delta}\}_k) = \frac{\sum_i d_i^{H_1}}{\sum_i (d_i^{H_1} + d_i^{\Delta})} \tag{12}$$

In the off chance that a number of malicious sensors from SSDF are positioned near each other, I want to have a level of tolerance z_θ and a required minimum number of sensors per cluster $C_{\min}$. The restraint $C_{\min}$ prevents a high clustering score z_k from an insignificant-sized cluster.

Figure 26 shows two scenarios: (1) the RSF-45 where each rogue antenna has a beamwidth of 45° and (2) the SSDF-40 where 40% of the sensors, randomly selected, perform SSDF. The red nodes are sensors reporting H_1, and the blue nodes are sensors reporting H_0. The red edges are formed when $c_i = c_j$ and $d_{ij} < d_\theta$ for sensors s_i and s_j. The blue edges are formed by the same rules except that $c_i \neq c_j$.



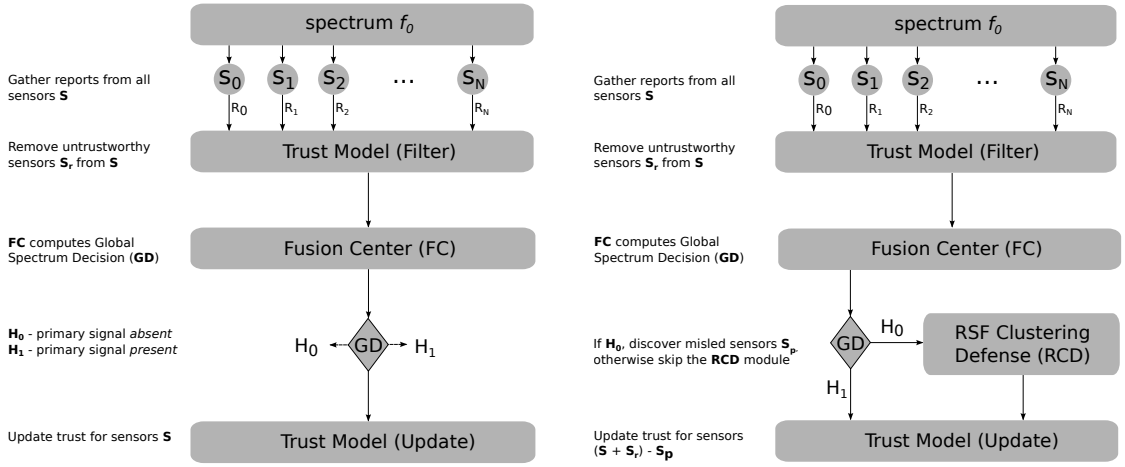
(a) RSF-45 (b) SSDF-40

Figure 26: Clustering illustration of my RSF Clustering Defense (RCD) algorithm. **(a)** RSF-45. **(b)** SSDF-40. The RCD forms two graphs, a red and blue graph, for cluster analysis. The red graph contains edges between sensors reporting H_1. The blue graph contains edges between sensors with opposing local spectrum decisions.

The red and blue graph both give valuable information in detecting directional rogue signals by the cluster formations they create. The goal of the red graph is to identify a strong concentration of sensors perceiving a radio signal within a small area. In contrast, the blue graph demonstrates disagreements in spectrum decisions (i.e., H_0 and H_1) between neighboring sensors. As can be seen in the RSF scenario in Figure 26a, the red graphs (created by the rogue signals) is surrounded by the blue graph without any significant overlap. The delineation between a red and blue graph roughly outlines a radio's antenna coverage and becomes a clear indication of a rogue signal. However, the SSDF scenario in Figure 26b shows that an overlapping of red and blue graphs reveal a strong likelihood of malicious or malfunctioning sensors, instead of a rogue signal's presence, since there is no apparent pattern of spectrum decisions (H_0, H_1).

5.2 Protocol and System Flow

My work revolves around a *centralized* trust-based CSS protocol, which means there is a dedicated base station that processes the cooperative spectrum sensing. This is contrasted with *local* trust-based CSS protocol which requires each cognitive radio to handle cooperative spectrum sensing on its own hardware. The benefit of having a centralized trust-based CSS protocol is: more feasible computing power to shoulder the burden of hosting Intrusion Detection Systems (IDS), such as the one I proposed.



(a) Conventional trust-based CSS protocol (b) RCD trust-based CSS protocol

Figure 27: Diagram of the trust-based CSS Protocol. Subfigure (b) adds the RCD module after the FC step, but only when the global decision GD = H_0.

Figure 31a illustrates the general framework of a trust-based CSS protocols [12, 30, 5]. In it, f_0 represents some wireless spectrum frequency, S_i a set of sensors, and R_i the corresponding set of sensor reports. The system model is a stack of dependent layers, starting with the spectrum channel, the network of sensors, the trust model, and finally the Fusion Center (FC). The accuracy of the CSS is dependent on the FC receiving reliable input from the above layers. For example, the spectrum channel must be clear enough for communication, the majority of sensors must not be malicious or malfunctioning, and the trust model must filter the malicious sensors to protect the

FC from bad input. The following list describes the variables in Fig 27:

- S - set of spectrum sensors (attached to cognitive radios)
- S_r - set of untrustworthy sensors flagged for removal by the trust model
- S_p - set of sensors protected by the RCD module
- H_0 - the null hypothesis that presumes the primary signal is present
- H_1 - the alternative hypothesis that presumes the primary signal is absent
- RCD - RSF (Rogue Signal Framing) Clustering Defense module

The RCD solution was designed to be modular, so that it could be inserted into existing trust-based CSS protocols as a panacea against RSF attacks. Figure 31b illustrates the order (after the fusion step) and the condition (FC reports global decision of H_0) for activating the RCD module. The intended goal was to prevent SUs from being penalized by rogue signals intended to make them look malicious. However, if an attack causes the FC to output a global decision of H_1, then that constitutes a PUE attack and requires a different solution altogether, such as the one presented in [14]. The following steps correspond to the trust-based CSS protocol with the RCD module from Figure 31b:

1. Collect all sensor reports from the network of sensors S
2. Apply the trust model's filter by removing untrustworthy sensors S_r from S
3. Make a global spectrum decision, denoted as GD, from sensors in $(S - S_r)$ as it normally would in trust-based CSS protocols
4. Discover signs of an RSF intrusion and identify the group of attacked sensors, denoted as S_P (for sensors protected)

5. Update the sensor reputations *except* for the set of sensors S_P that are presumed affected by rogue signals

5.3 Overhead of Defense

To address the time complexity overhead of my defense, I have to examine the algorithm it uses before I can identify the order-of-growth category it belongs to. The proposed RSF Clustering Defense (RCD) algorithm can be separated into three distinct parts; (1) the graph setup, (2) the Breadth-First-Search to identify all the clusters (i.e. subgraphs), and (3) calculating the clustering strength of an identified cluster. Each part can be summarized by the following:

1. Connect all the vertices in the adjacency matrix A_{ij} to its neighbors within a distance threshold d_θ; this step has a time complexity of $O(|V|^2)$ where $|V|$ is the number of sensors

2. Find all non-overlapping subgraphs (i.e. clusters C_k) using a Breadth-First-Search; this step has a time complexity of $O(|V|^2)$ since it traverses the adjacency matrix A_{ij} and creates adjacency lists that represent each C_k cluster

3. Calculate the clustering strength of cluster C_k based on the assortative mixing equations (eq 9 and eq 10); this step iterates through each C_k adjacency list, thus it has a time complexity of $O(|E| + |V|)$

So the time complexity of the RCD defense is the summation of all three parts: $O(|V|^2) + O(|V|^2) + O(|E| + |V|)$. Yet, in a static network, where the cognitive radios do not move, I can ignore the complexity of part 1 since it is only computed once during the program initialization. Hence, the time complexity for each reoccurring quiet period is $O(|V|^2) + O(|E| + |V|)$. The quiet period is when the cognitive radio network stops transmitting to listen for the primary signal.

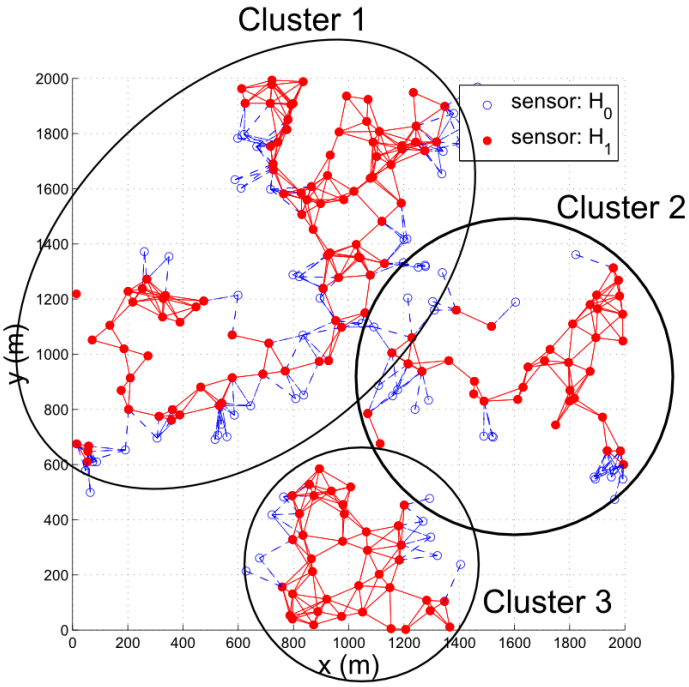


Figure 28: The sensor network is partitioned into a red and blue graph before being analyzed by the RCD module. The red filled nodes are cognitive radios reporting H_1 and are connected to nearby neighbors with similar observations.

The bottleneck of my defense is either in part 2 or part 3, whichever has a worse order of growth between $O(|V|^2)$ and $O(|E| + |V|)$, depending on the sizes of V and E. The RCD algorihm travereses through K adjacency lists representing each cluster C_k, where $0 \leq k < K$. Fig. 28 shows $K = 3$ clusters present (C_0, C_1, and C_2) in the network where each cluster is roughly 1/4 to 1/8 the size of V.

Time complexity can be an issue if an attack is able to impact the network before the defense can adequately prevent or mitigate the damage. However, my algorithm has a descent order of growth, i.e. $O(|V|^2) + O(|E| + |V|) \approx O(|V|^2)$, which is smaller than many clustering algorithms such as the Kernighan-Lin algorithm that have an order of growth of $O(|V|^3)$. Secondly, I assume that all intensive processing happens at the base station, with a dedicated server and adequate computing resources performing the analysis, and not on the cognitive radios itself. As such, the time complexity is

very feasible for most anticipated network sizes, e.g. no more than several thousand sensors. Furthermore, the calculation of the clustering strength is only applied to small sections of the network, which is usually much smaller than the total number of sensors $|V|$. This occurs in part 2 of my defense where C_k clusters with identical sensor reports are identified using BFS, in similar fashion to the Flood Fill algorithm.

The need for more intensive processing, like graph algorithms, in radio networks usually raises concerns about the impact it has on a radio's battery life. This is not a concern in my system, because the cognitive radios only submit sensor reports every 30 seconds to a stationary base station that does all the processing on a dedicated server. Hence, the cognitive radios are spared the processing that would otherwise quickly deplete itself of battery life. In a decentralized CSS protocol, each cognitive radio is responsible for computing the shared spectrum algorithms locally, but my system employs a centralized CSS protocol which removes the intensive processing burden on the radio itself.

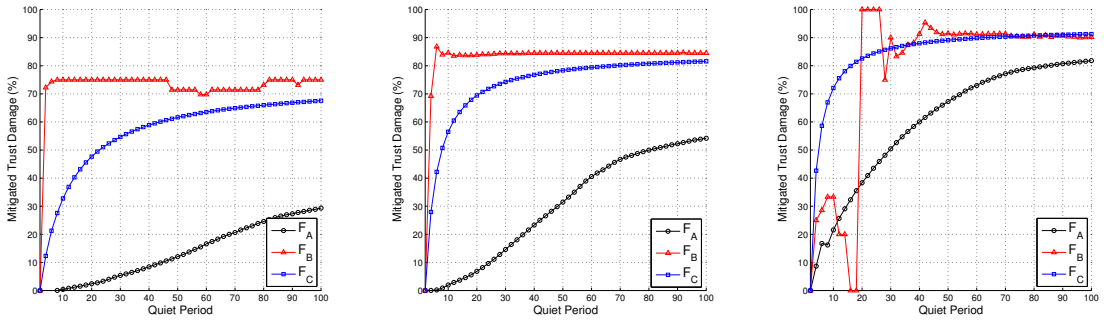
5.4 Defense Evaluation

In this section, I evaluate the RCD module's performance on its ability to mitigate trust loss from RSF intrusions. Additionally, I compare the RCD module's outcome on RSF and SSDF attacks.

In my simulations, I have two groups of scenarios, the RSF and SSDF. The simulation environment is the same as the one used by the RSF intrusion in Section 4. The beamwidth of each rogue antenna is 15°, 30°, and 45° for scenarios RSF-15, RSF-30, and RSF-45, respectively. The SSDF scenarios simulate malicious sensors by randomly selecting a percentage of the sensors and raising their RSS by 20 dBm from the noise floor. I randomly selected 20%, 30%, and 40% of sensors from the scenarios SSDF-20, SSDF-30, and SSDF-40, respectively.

Fig. 29 shows the amount of mitigated trust damage (%) with the RCD module under the same scenarios. The mitigated trust damage is denoted as $T_M[q]$ and calculated by:

$$T_M[q] = \frac{T_\Sigma^R[q] - T_\Sigma[q]}{T_\Sigma[0] - T_\Sigma[q]} \tag{13}$$



(a) RSF-15 (b) RSF-30 (c) RSF-45

Figure 29: Displays the network's total mitigated trust damage (defined in Eq. 13) from the RCD module.

where $T_\Sigma^R[q]$ is the network's total trust on quiet period q when using the RCD module, $T_\Sigma[q]$ is the network's total trust without the RCD module (from Fig. 17), and $T_\Sigma[0]$ is the initial state of trust scores. I use a minimum cluster size $C_{min} = 5$, a clustering threshold $Z_\theta = 0.3$, and a distance threshold $d_\theta = 150$ m.

As shown in Fig. 29, each protocol benefited from my proposed defense against the RSF intrusion. However, the RCD module offered less protection to protocol F_A due its sequential random sampling of sensors, instead of cross-examining all sensor reports for a more robust analysis. The spikes from F_B in Fig. 29c) are due to its protocol design of having a dynamic threshold for deciding malicious sensors. During the spikes, F_B's dynamic threshold is stabilizing as it replaces the old RSS statistics with new data.

Fig. 30 shows the rate of false alarms, i.e. the number sensors reporting H_1 when

the FC reports H_0, before and after applying the RCD module. In all three RSF scenarios, the RCD module managed to limit the false alarms to a maximum of 3% of total sensors N_s.

Fig. 31 compares how the RCD responds to the RSF and SSDF intrusions in terms of the number of sensors attacked S_A and the number of sensors protected S_P by the RCD module. The goal is to maximize S_P for the RSF scenarios and minimize it for the SSDF scenarios so that the reputations of malicious sensors are not protected. In scenario RSF-45, the strongest RSF attack, the RCD module protects 95% of sensors from losing trust due to rogue signals. In contrast, the RCD module erroneously protects 15% of the sensors in scenario SSDF-40. This margin of error is acceptable as 40% of malicious sensors is an unrealistic and profuse amount of attacks in any CR network. The outcomes of Fig. 31 show a high resiliency against the exploitation of SSDF attacks.

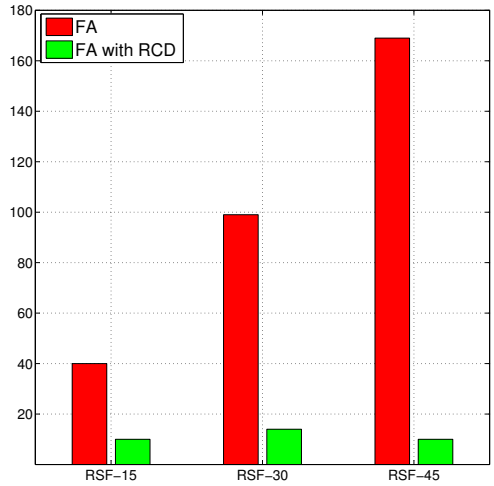
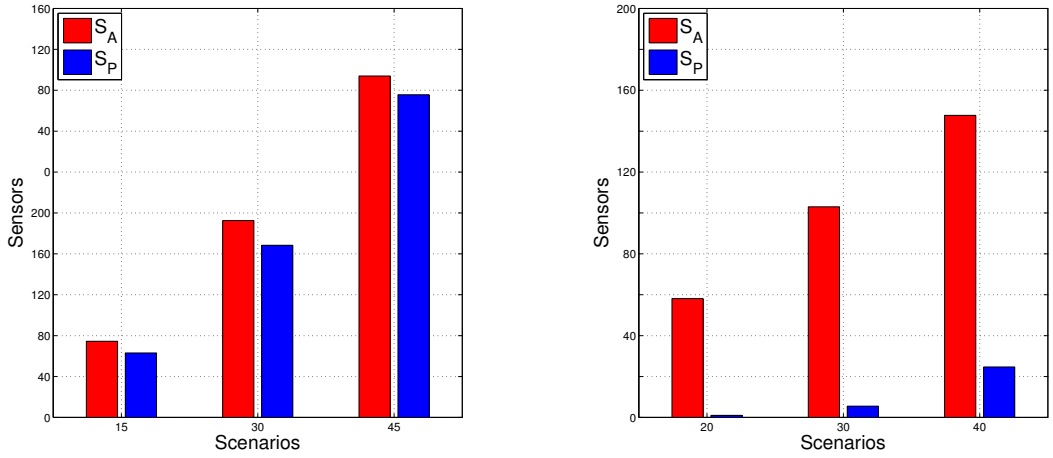


Figure 30: The number of false alarms before and after applying the RCD module.



(a) Conventional trust-based CSS protocol (b) RCD trust-based CSS protocol

Figure 31: Comparison of the RCD results between RSF and SSDF intrusions. S_A - number of attacked sensors; S_P - number of sensors protected by the RCD

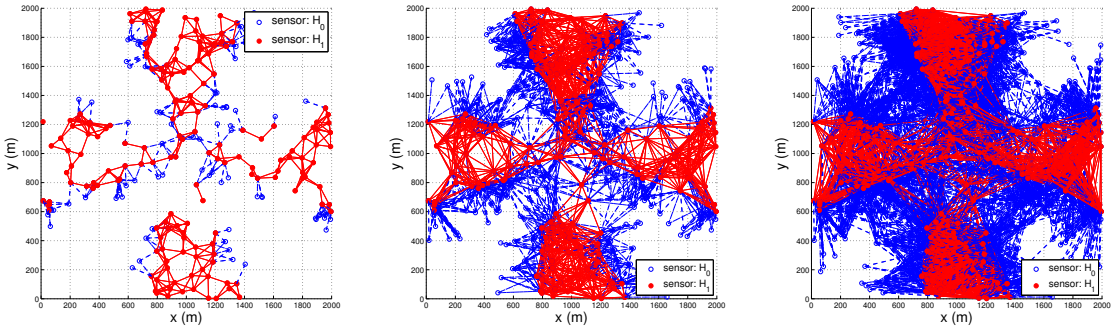
5.5 Cluster Parameters and Impact

Naturally, the size and topology of the cognitive radio network has an affect on the RCD solution. A dense network can easily show patterns of rogue signals where as a sparse network gives less information to analyze. To show the difference, I tested my solution on a second network, denoted as the sparse network, consisting of 100 randomly placed sensors. In contrast, the dense network has 400 randomly placed sensors, which is the same network tested and discussed in previous sections. For both dense and sparse networks, I only display the RSF-45 scenario to limit the number of graphs. The RSF-45 scenario emits four rogue signals in the cardinal directions with 45° beamwidth.

The distance threshold d_θ is the condition required to form edges between two sensors. A red graph indicates a strong concentration of sensors perceiving a signal, such that it potentially reveals a rogue signal's antenna coverage. The red graph is formed by sensors that share H_1 reports within the distance threshold, d_θ. Likewise,

the blue graph is formed by sensors that simply disagree with their neighbors' spectrum decisions (i.e. H_0 and H_1) within d_θ. The blue graph helps reveal an SSDF attack, especially when the red and blue graph are overlapping, and not clearly segregated. When a rogue signal is present, the red graph should be surrounded by the blue graph, outlining the reach of the rogue signal's antenna coverage.

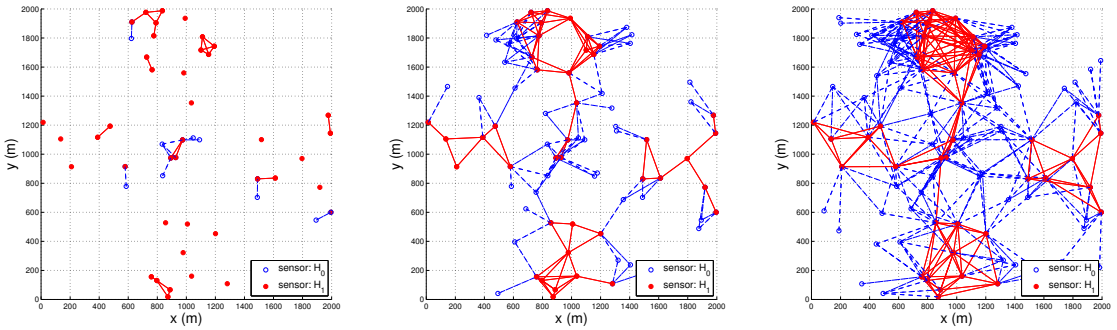
Fig. 32 and Fig. 33 show the changing composition in the red and blue graph (created by the RCD) in both dense and sparse networks with different d_θ, where $d_\theta = 150, 300, 450$ m. For the dense network, the attack coverage of the rogue signals is clearly visible with all three values for d_θ. For the sparse network, the visibility of rogue signals becomes much more difficult to perceive, especially when $d_\theta = 150$ m. Naturally, this occurs from having fewer sensors, randomly placed, over the same area as the dense network. In other words, the sensors are farther away from their neighbors in the sparse network.



(a) RSF-15 (b) RSF-30 (c) RSF-45

Figure 32: RCD solution applied to a dense network of 400 sensors.

At first glance, it might be tempting to just assign an excessive number for d_θ to avoid the sparsity problem, i.e. when clusters are not clearly visible because d_θ is too low. Actually, a very large d_θ can decrease the accuracy of the RCD solution as shown in Fig. 34. An infinitely large d_θ will always form complete blue and red graphs across



(a) RSF-15 (b) RSF-30 (c) RSF-45

Figure 33: RCD solution applied to a sparse network of 100 sensors.

the sensor region, which is not always more informative.

Fig. 34 shows the accuracy of the RCD solution for both dense and sparse networks with $d_\theta = 150, 300, 450$ m. The accuracy is represented by the *number of sensors protected by the RCD solution* divided by the *number of sensors inside the rogue signal's attack coverage*, i.e. S_P/S_A. Notably, the $d_\theta = 300$ m in the sparse network reaches 100% accuracy, but $d_\theta = 450$m does not, even with more edges to analyze. The reason for this phenomena is due to the blue edges lowering the clustering score Z_k for cluster C_k. This can seen in eq. 12, where the clustering score Z_k decreases because the denominator increases as more blue edges form (from variable d_i^Δ).

There are many variables in the simulations that are worth analyzing at a more comprehensive level. The number of sensors, the number of attackers, the shape and size of the rogue signal, the network's topology, and even the environment's landscape.

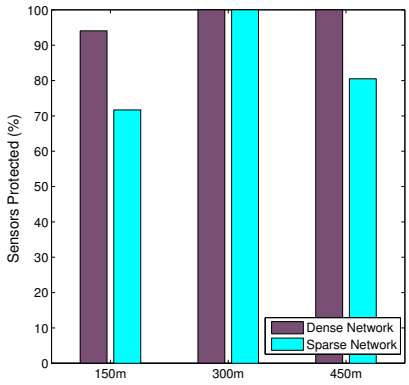


Figure 34: The accuracy of the RCD for dense and sparse networks with $d_\theta = 150, 300, 450$m.

6 Dynamic Clustering Methods

In this chapter, I evaluate the RCD modules performance on its ability to detect rogue signals in several scenarios with different parameters, which include RSF attack coverage and network size. Additionally, I compare the RCD modules outcome on RSF and SSDF attacks to evaluate its robustness.

6.1 Clustering Methods

There are many variables to consider in the context of detecting rogue signals through CSS. The number of sensors, the number of attackers, the beamwidth and transmission power of the rogue signal, the networks topology, and even the environments landscape (that accounts for shadow fading). Thus, I look toward a solution that dynamically adjusts according to the network size and density. In short, I aimed to devise a parameter-free algorithm that 1) was effective in all cases and 2) the solution did not require endless tweaks for optimal results.

I discuss three clustering methods to detect rogue signals, built on top of the RCD foundation of the previous subsection. The first method, Static Distance Threshold

(SDT), was named as such since the distance threshold d_θ (threshold that determines if an edge exists between sensors s_i, s_j) is statically and arbitrarily chosen. The other two methods, K Nearest Neighbors (KNN) and Median Distance Threshold (MDT), have thresholds that dynamically change according to the network's size and density at a given location, and pertains to this paper's contribution. The three methods are explained as follows:

1. **Static Distance Threshold (SDT)**

   Edges are formed between any two sensors, s_i and s_j, if the distance between the two is below the distance threshold d_θ, which eventually culminates into a cluster. The defining characteristic of the SDT method is the arbitrary and static assignment of d_θ. The downside was having to arbitrarily determine d_θ which may or may not be effective in a given scenario.

2. **K Nearest Neighbors (KNN)**

   For the KNN-inspired method, the distance threshold is dynamically chosen as $d_\theta = d_{ik}$, the distance between s_i and the k^{th} closest sensor s_k. In short, this method forms up to k edges for sensor s_i, but only when s_i reports H_1, *i.e,* it perceives the primary signal. The downside is that k must be arbitrarily chosen, and it is not apparent which is the best k for a given scenario.

3. **Median Distance Threshold (MDT)**

   As the name implies, the distance threshold $d_\theta = median_i$ where $median_i$ is the median distance of all the distances d_{ij} between s_i and s_j in the immediate or adjacent grid units. Like SDT, edges are formed between any two sensors, s_i and s_j, if $d_{ij} \leq d_\theta$ and s_i reports H_1. The proposed scheme does not need to set a subjective threshold d_θ or rely on the estimation of the network's density, which means that the method is robust to sensor locality.

However, in all three methods, the clustering strength threshold z_θ for cluster $C_k \in S$ has to be manually set. The clustering strength z_θ corresponds to the solution's sensitivity in locating a rogue signal; smaller z_θ leads to more misdetections and a larger z_θ leads to more false alarms.

6.2 Clustering Threshold Determined by Locality

The RSF defense needs to cross examine a sensor s_i with all other sensors in S to see if it met the conditions for creating an edge, e.g., if the distance d_{ij} between s_i and s_j was $d_{ij} \leq d_\theta$. To improve both performance and efficiency, edges between sensors s_i, s_j are considered only if sensor s_j is in the immediate or adjacent cell.

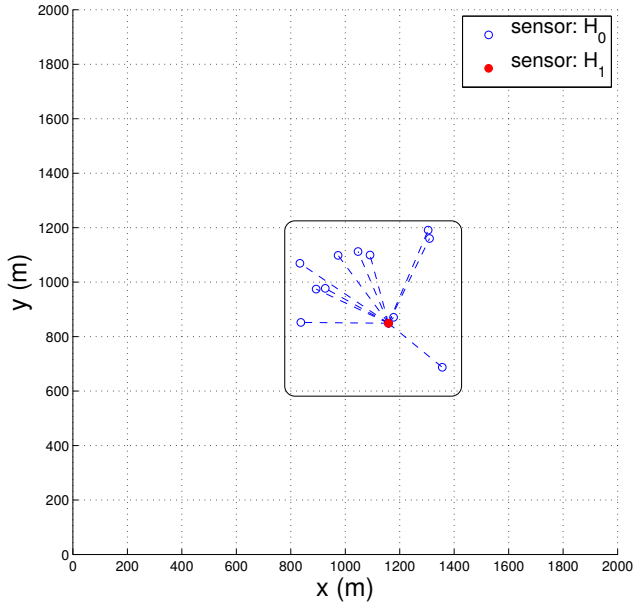


Figure 35: To improve efficiency, edges between sensors s_i, s_j are considered only if sensor s_j is in the immediate or adjacent cell

The RCD algorithm was modified to only form edges with sensors in the immediate or adjacent grid units, with the use of hashmaps, in order to mitigate the overhead of

forming edges between all sensors. In Figure 35, a sensor reporting H_1 (primary signal present) forms 11 edges with neighboring sensors reporting H_0 (primary signal absent) in the immediate or adjacent grid cells. In my example, the red sensor reporting H_1 is connected to 11 blue edges, which means all of its neighbors are reporting differently, i.e., H_0. The optimization relies on a hashmap (*key,value*) structure where the *key* is the grid cell id (e.g. row and column id) and the *value* is the set of sensors in that grid cell. There is no need to check for edges after a certain distance, since the further two sensors are apart, the weaker the RSS correlation [35]. This optimization alleviates the computation burden of the clustering algorithm. In my example, the size of a single grid cell is 200×200 in a grid of size 2000×2000.

Secondly, and more importantly, the clustering threshold d_θ differs from one area to another within the grid space, that allows the solution to conform to different community densities in a large space. Take, for example, a grid cell and its neighboring cells, together denoted as G_1, that has a $d_\theta = 200$ determined by the median distance of sensors S_1 in G_1. Another grid cell and its neighbor cells, denoted as G_2, might have a $d_\theta = 300$ instead. Thus we have n different $d_{i\theta}$ for each G_i where n is the number of grid cells and $0 < i \leq n$. Having n different clustering thresholds is exactly gives the solution the parameter-free property. The clustering methods MDT and KNN simply determine the local distance thresholds $d_{i\theta}$ for each G_i.

6.3 Simulation Setup

I have two types of networks for my simulations, a dense network of size 400 sensors and sparse network of size 100, which are located inside a $2,000 \times 2,000$ grid. I assume the incumbent broadcasting station operates at the UHF frequency of 615 MHz. Like Figure 12, there are four rogue directional antennas facing the cardinal directions and positioned on the map's center. I created three RSF attack scenarios, labeled as RSF-

15, RSF-30, and RSF-45 which corresponds to the scenario's antenna beamwidths of 15°, 30°, and 45°, respectively. The SSDF scenarios simulate malicious sensors by randomly selecting a percentage of the sensors and raising their RSS by 20 dBm from the noise floor. I randomly selected 20%, 30%, and 40% of sensors from the scenarios SSDF-20, SSDF-30, and SSDF-40, respectively.

Table 8: Scenario Types

LABEL	ATTACK	DESCRIPTION
RSF-15	Rogue Signal Framing	4 antennas, beamwidths=15°
RSF-30	Rogue Signal Framing	4 antennas, beamwidths=30°
RSF-45	Rogue Signal Framing	4 antennas, beamwidths=45°
SSDF-10	Spectrum Sensing Data Falsification	10% of sensor reports falsified
SSDF-20	Spectrum Sensing Data Falsification	20% of sensor reports falsified
SSDF-30	Spectrum Sensing Data Falsification	30% of sensor reports falsified

Table 8 displays all the attack scenarios I simulated to test my proposed RSF defense.

6.4 Comparison of Clustering Methods

This subsection displays the test results of the three clustering-based rogue signal detection methods (SDT, KNN, MDT) against the attack scenarios listed in Table 8.

There are two types of edges in my cluster network; red edges which are pairs of sensors reporting H_1 and blue edges which are pairs reporting differently. The clustering strength z_θ is the purity of a cluster community, so 0.33 means at least a third are red edges. I picked a clustering strength of 0.33 because it proved to be very robust to different network sizes, along with other values between 0.3 and 0.5. Thresholds higher than 0.5 led to many rogue signals going unnoticed (misdetection), and anything lower than 0.3 resulted in SSDF attacks being confused for RSF attacks (false alarms).

For the KNN tests, I set K=10. Table 9 lists the results of the three clustering-based rogue signal detection methods on a dense network of size 400, while Table 10 lists the corresponding results on a sparse network of size 100. The table values are fractions where the numerator is number of sensors protected by the RCD (denoted as S_p) and the denominator is the number of sensors affected by the rogue signal (denoted as S_a). For the RSF columns, a higher percentage means a better detection rate. For the SSDF columns, a smaller percentage (which is better) means less false alarms from an SSDF attack.

Table 9: Performance of the three clustering methods in a dense network of size 400, in the form of S_p/S_a (number of sensors protected over sensors attacked)

	RSF-15	**RSF-30**	**RSF-45**	**SSDF-10**	**SSDF-20**	**SSDF-30**
SDT	35/40	95/99	159/169	3/40	16/80	52/120
KNN	35/40	99/99	169/169	0/40	0/80	46/120
MDT	20/40	99/99	169/169	0/40	0/80	15/120

Table 10: Performance of the three clustering methods in a sparse network of size 100, in the form of S_p/S_a (number of sensors protected over sensors attacked)

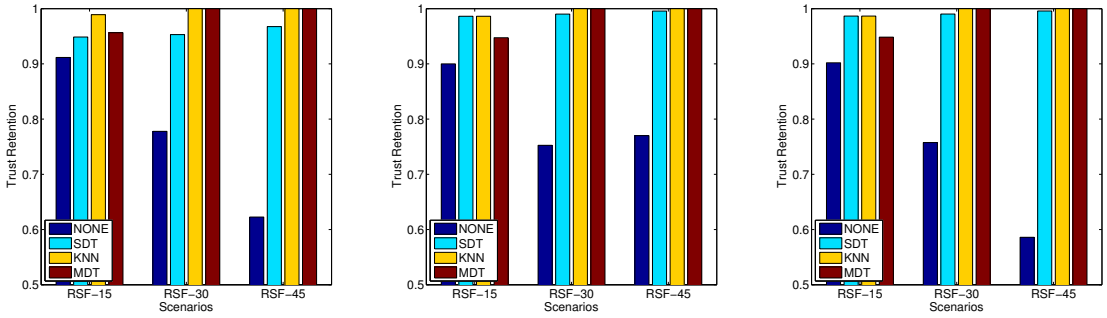
	RSF-15	**RSF-30**	**RSF-45**	**SSDF-10**	**SSDF-20**	**SSDF-30**
SDT	2/11	9/25	24/41	0/10	5/20	7/30
KNN	5/11	20/25	41/41	0/10	0/20	5/30
MDT	7/11	22/25	40/41	0/10	0/20	0/30

The MDT and KNN methods easily outperform the SDT method in both types of attacks (RSF and SSDF). Interestingly, the MDT and KNN are virtually tied in performance, with KNN outperforming in some cases but not always. However, MDT appears to have more resilience against SSDF attacks, meaning that it can better distinguish between RSF and SSDF attacks, i.e. between rogue signals and malicious SUs. It is worth noting that the MDT method is also preferable because the KNN method requires an arbitrarily chosen K value, which may or may not be optimal for the given scenario.

6.5 RSF Defense on Trust-based CSS Protocols

I apply clustering methods SDT, KNN, and MDT to three different trust-based CSS protocols (which I borrow from [12, 30, 5]) to analyze my solution's ability to protect sensor reputations. The previous subsection simply compares the accuracy of detecting sensors affected by rogue signals, which is represented by (S_p/S_a).

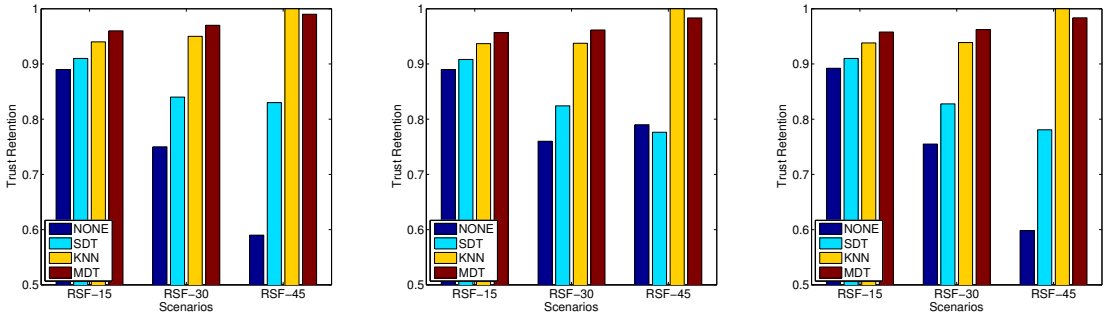
The first protocol F_A, by Chen et al. [12], utilizes the sequential probability ratio test (SPRT) and weights the probability by the sensor's reputation to mitigate the impact of SSDF attacks. The second protocol F_B, by Kaligineedi et al. [30], utilizes a pre-filtering average combination scheme. These filters are responsible for (1) filtering extreme outlier sensor reports and (2) ignoring sensors with high-trust penalties. The third protocol F_C, by Arshad et al. [5], utilizes a beta reputation system model for hard-decision CSS protocols. Like F_A, the sensors are rewarded for agreeing with the global spectrum decision, but otherwise penalized.



(a) F_A (b) F_B (c) F_C

Figure 36: Comparison of clustering techniques with protocols F_A, F_B, and F_C on a dense network.

Figures 36 and 37 shows the network's reputation retention (i.e., the system's total trust on a scale between 0 and 1) at the end of a sustained RSF attack lasting 50 quiet periods. Besides incorporating the RSF defense into the three protocols, the parameters are the same from the "Simulation Setup" subsection.



(a) F_A (b) F_B (c) F_C

Figure 37: Comparison of clustering techniques with protocols F_A, F_B, and F_C on a sparse network.

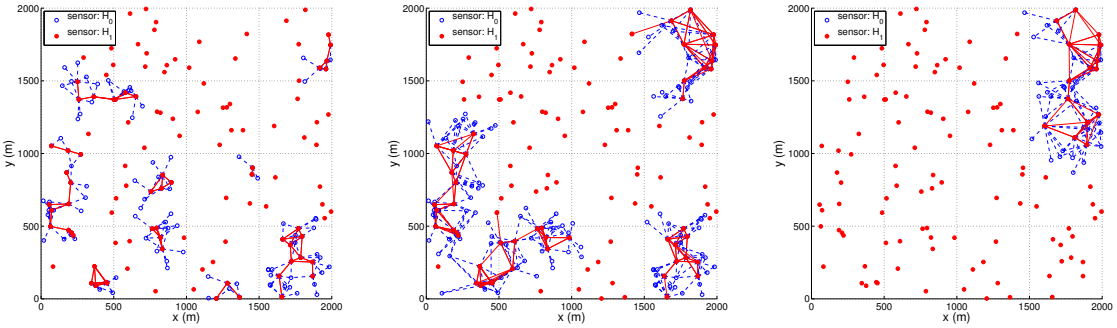
As expected, Figure 36 shows all clustering methods providing nearly perfect protection of sensor reputations. The experiment's parameters, along with the SDT method, were used in my previous work [28], and showed similar results. However, Figure 37 shows KNN and MDT methods clearly outperforming the SDT method in sparse networks. In particular, the KNN and MDT methods perform roughly 10% better in RSF-30 scenarios and 20% better in RSF-45 scenarios. Simply put, the SDT method is not flexible or robust enough to handle different network densities, especially when the distance threshold is statically assigned.

I purposely picked a distance threshold $d_\theta = 150$ to show that clustering parameters must be dynamic to account for certain variables in an attack scenario, such as the proximity of sensors in cooperative spectrum sensing. As indicated in my test results, it worked well for dense networks, but not in sparse networks where the average distance between sensors was greater.

6.6 Clustering Figures - False Alarms (SSDF)

Figure 38 illustrates three clustering methods (SDT, KNN, MDT) on a dense network of size 400 in a $2,000 \times 2,000$ grid, and a clustering threshold $z_\theta = 0.33$. For the SDT

method, I used a distance threshold of $d_\theta = 150m$. The purpose of these figures is to test the resilience of the RCD solution against large scale SSDF attacks. In particular, I am testing the RCD module's ability to distinguish between RSF and SSDF attacks. The false alarm rate corresponds (roughly) to the number of edges in a figure, so less edges is better in this case. The exact number of false alarms is in the SSDF-30 column in Table 9.



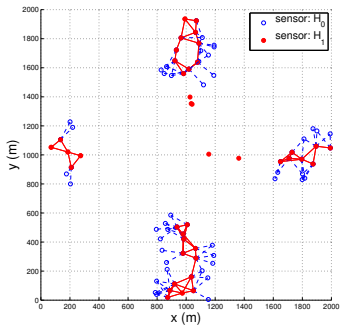
(a) SDT (b) KNN (c) MDT

Figure 38: Demonstrates the clustering behavior of SDT, KNN, and MDT methods on the SSDF-30 scenario applied to a dense network, or simply put, when 120 out of 400 sensors suffer an SSDF attack

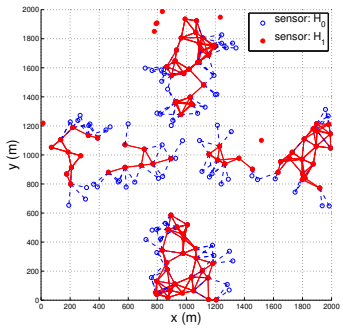
As demonstrated in Figure 38, the MDT method appears to have the least amount of false alarms, meaning that it can better distinguish between RSF and SSDF attacks, i.e. between rogue signals and malicious SUs. It is worth noting that the MDT method is also preferable because the KNN method requires an arbitrarily chosen K value, which may or may not be optimal for the given scenario.

6.7 Clustering Figures - Dense Network

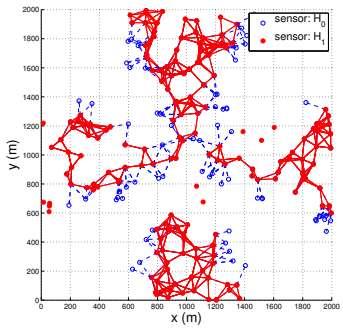
This subsection presents figures to illustrate the three clustering methods (SDT, KNN, MDT) on a dense network of size 400 in a $2,000 \times 2,000$ grid. These figures reflect the RSF columns from Table 9. For the SDT method, I used a distance threshold of $d_\theta = 150m$. As illustrated in Figure 12, the rogue transmitters are positioned at the center of the map.



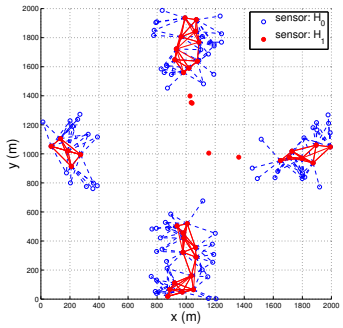
(a) RSF-15, SDT



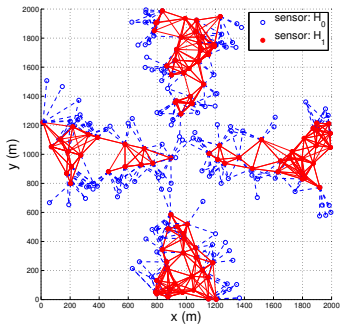
(b) RSF-30, SDT



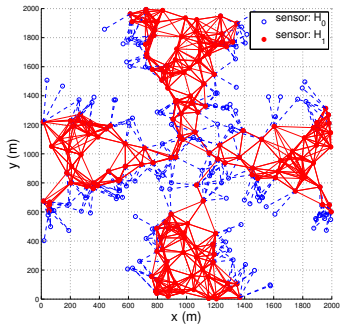
(c) RSF-45, SDT



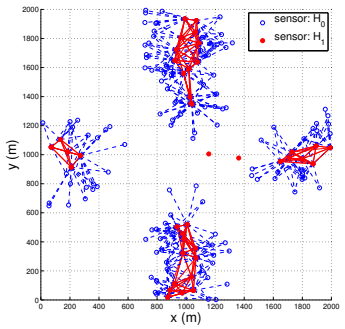
(a) RSF-15, KNN



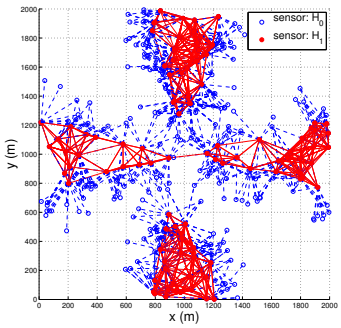
(b) RSF-30, KNN



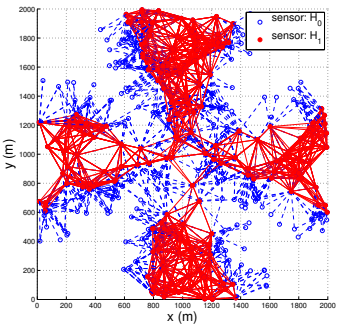
(c) RSF-45, KNN



(a) RSF-15, MDT



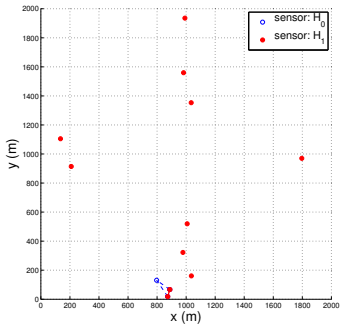
(b) RSF-30, MDT



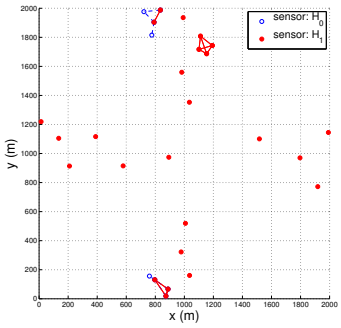
(c) RSF-45, MDT

6.8 Clustering Figures - Sparse Network

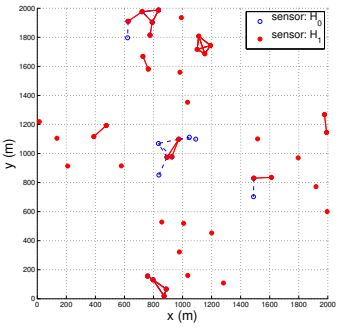
This subsection presents figures to illustrate the three clustering methods (SDT, KNN, MDT) on a sparse network of size 100 in a $2,000 \times 2,000$ grid. These figures reflect the RSF columns from Table 10. For the SDT method, I used a distance threshold of $d_\theta = 150m$. As illustrated in Figure 12, the rogue transmitters are positioned at the center of the map.



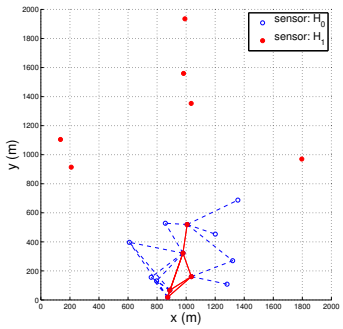
(a) RSF-15, SDT



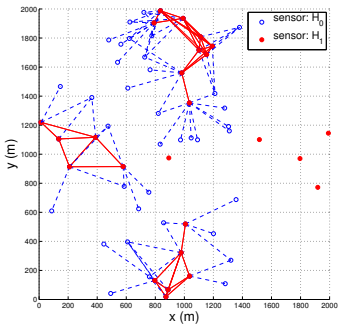
(b) RSF-30, SDT



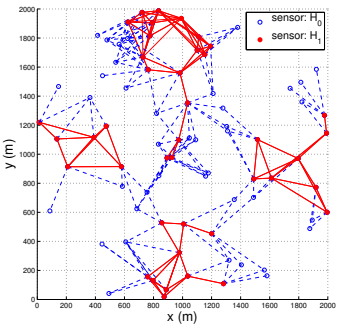
(c) RSF-45, SDT



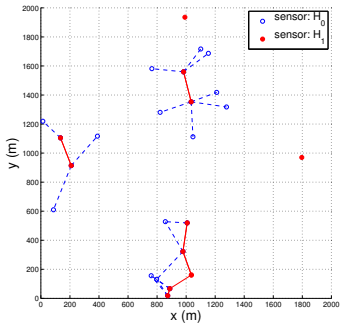
(a) RSF-15, KNN



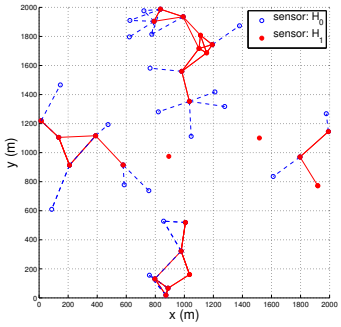
(b) RSF-30, KNN



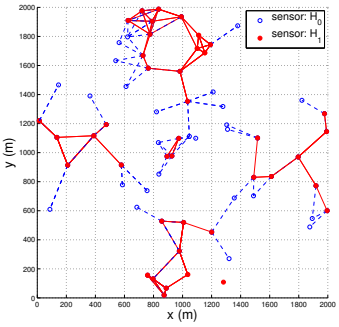
(c) RSF-45, KNN



(a) RSF-15, MDT



(b) RSF-30, MDT



(c) RSF-45, MDT

7 Conclusion

My work demonstrates the RSF intrusion, a new threat to trust-based CSS protocols. The attackers can transmit rogue signals onto groups of sensors to emulate SSDF and ruin their reputation with the intent of having them removed from the shared spectrum sensing. My findings caution the use of trust-based CSS protocols and warrants a line of defense against rogue signals. The RSF simulations were conducted in a realistic environment based on the 802.22 WRAN standard and illustrates the impact of the RSF intrusions on sensor reputation scores. To mitigate the trust damage, I introduced a new defense based on community detection via cluster analysis. The simulation experiments showed that my defense solution, the RCD module, could effectively keep the sensor reputations intact while distinguishing rogue signals from malicious sensors.

I discuss the challenges in detecting rogue signals in cooperative spectrum sensing, in an effort to better mitigate the impact of RSF attacks on sensor reputations. Additionally, my rogue signal detection solution has a dynamic clustering threshold based on the density of the network at a given location. This gives the advantage of a one-size-fits-all solution when it comes to handling networks that are sparse, dense, and disproportionate. My work contributes to making cognitive radio networks a viable technology, in particular, promoting research that helps shed some light on the difficulties of utilizing fallow spectrum safely.

References

[1] CRAWDAD - A Community Resource for Archiving Wireless Data At Dartmouth. `http://crawdad.org/`. [Online; accessed 27-Apr-2013].

[2] A. D. AIR-4.5. *Electronic Warfare and Radar Systems Engineering Handbook*. Naval Air Systems Command, 2006.

[3] M. Akbari and A. Falahati. Ssdf protection in cooperative spectrum sensing employing a computational trust evaluation algorithm. In *Proc. of the 5th International Symposium on Telecommunications 2010 (IST 2010), Tehran, Iran*, pages 23–28, Dec 2010.

[4] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50(13):2127–2159, 2006.

[5] K. Arshad and K. Moessner. Robust collaborative spectrum sensing based on beta reputation system. In *Proc. of the 20th Future Network Mobile Summit 2011 (FutureNetw 2011), Warsaw, Poland*, pages 1–8, June 2011.

[6] K. Bauer, D. McCoy, E. Anderson, M. Breitenbach, G. Grudic, D. Grunwald, and D. Sicker. The directional attack on wireless localization -or- how to spoof your location with a tin can. In *Proc. of the IEEE Global Telecommunications Conference, 2009 (GLOBECOM 2009), Honolulu, Hawaii, USA*, pages 1 –6. IEEE, 30 2009-Dec. 4 2009.

[7] S. Bhattacharjee, S. Debroy, and M. Chatterjee. Trust computation through anomaly monitoring in distributed cognitive radio networks. In *Proc. of the IEEE*

22nd International Symposium on Personal Indoor and Mobile Radio Communications 2011 (PIMRC '11), Toronto, Canada, pages 593 –597. IEEE, Sept. 2011.

[8] K. Borle, B. Chen, and W. Du. A physical layer authentication scheme for countering primary user emulation attack. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 2935–2939, May 2013.

[9] J. Burbank. Security in cognitive radio networks: The required evolution in approaches to wireless network security. In *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, pages 1 –7, may 2008.

[10] K. Chauhan and A. Sanger. Survey of security threats and attacks in cognitive radio networks. In *Electronics and Communication Systems (ICECS), 2014 International Conference on*, pages 1–5, Feb 2014.

[11] R. Chen and J.-M. Park. Ensuring trustworthy spectrum sensing in cognitive radio networks. In *Proc. of the 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks, 2006 (SDR '06), Reston, Virginia, USA*, pages 110 –119. IEEE, sept. 2006.

[12] R. Chen, J.-M. Park, and K. Bian. Robust distributed spectrum sensing in cognitive radio networks. In *Proc. of the 27th IEEE International Conference on Computer Communications 2008 (INFOCOM 2008), Phoenix, Arizona, USA*, pages –. IEEE, April 2008.

[13] R. Chen, J.-M. Park, Y. Hou, and J. Reed. Toward secure distributed spectrum sensing in cognitive radio networks. *IEEE Communications Magazine*, 46(4):50 –55, April 2008.

[14] R. Chen, J.-M. Park, and J. Reed. Defense against primary user emulation attacks in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 26(1):25–37, Jan 2008.

[15] T. Clancy and N. Goergen. Security in cognitive radio networks: Threats and mitigation. In *Proc. of the 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008 (CrownCom 2008), Singapore*, pages 1 –8, may 2008.

[16] F. C. Commission. Second memorandum opinion and order, in the matter of unlicensed operation in the tv broadcast bands, additional spectrum for unlicensed devices below 900 mhz and in the 3 ghz band, September 2010. Document 10-174.

[17] R. Etkin, A. Parekh, and D. Tse. Spectrum sharing for unlicensed bands. *Selected Areas in Communications, IEEE Journal on*, 25(3):517–528, April 2007.

[18] FCC. Public Safety Tech Topic #9 - Cognitive Radio Potential for Public Safety. `https://www.fcc.gov/help/public-safety-tech-topic-9-cognitive-radio-potential-public-safety`, 2015. Accessed: 2015-03-07.

[19] Forbes. A Simple Explanation Of 'The Internet Of Things'. `http://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/`, 2014. Accessed: 2015-03-011.

[20] I. Forkel, M. Schinnenburg, and M. Ang. Generation of two-dimensional correlated shadowing for mobile radio network simulation. In *Proc. of the 7th International Symposium on Wireless Personal Multimedia Communications 2004*

(WPMC 2004), Sydney, Australia, page 5, Abano Terme (Padova), Italy, Sep 2004.

[21] A. Fragkiadakis, E. Tragos, and I. Askoxylakis. A survey on security threats and detection techniques in cognitive radio networks. *Communications Surveys Tutorials, IEEE*, 15(1):428–445, First 2013.

[22] A. Gorcin and H. Arslan. Public safety and emergency case communications: Opportunities from the aspect of cognitive radio. In *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pages 1–10, Oct 2008.

[23] L. Guijarro, V. Pla, and J. Vidal. Competition in cognitive radio networks: Spectrum leasing and innovation. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 1112–1113, Jan 2011.

[24] S. Higginbotham. Spectrum Shortage Will Strike in 2013 Tech News and Analysis. `http://gigaom.com/2010/02/17/analyst-spectrum-shortage-will-strike-in-2013/`, 2013. [Online; accessed 08-Apr-2013].

[25] C. Huifang, X. Lei, and N. Xiong. Reputation-based hierarchically cooperative spectrum sensing scheme in cognitive radio networks. *Communications, China*, 11(1):12–25, Jan 2014.

[26] IEEE. Ieee starts standard to tap open regions in the tv spectrum for wireless broadband services. `http://web.archive.org/web/20090207021748/http://standards.ieee.org/announcements/pr_80222.html`, 2004. Accessed: 2015-03-14.

[27] G. Inc. Spectrum Database. `https://www.google.com/get/spectrumdatabase/channel/`, 2015. Accessed: 2015-03-14.

[28] D. S. Jackson, W. Zang, Q. Gu, W. Cheng, and M. Yu. Exploiting and defending trust models in cooperative spectrum sensing. *EURASIP Journal on Wireless Communications and Networking 2015*, January 2015.

[29] T. Jing, X. Chen, Y. Huo, and X. Cheng. Achievable transmission capacity of cognitive mesh networks with different media access control. In *Proc. of the 31nd IEEE International Conference on Computer Communications (INFOCOM 2012), Orlando, Florida, USA*, pages 1764–1772, March 25-30 2012.

[30] P. Kaligineedi, M. Khabbazian, and V. Bhargava. Secure cooperative sensing techniques for cognitive radio systems. In *Proceedings of IEEE International Conference on Communications 2008 (ICC '08), Beijing, China*, pages 3406–3410. IEEE, May 2008.

[31] A. Kuntz, F. Schmidt-Eisenlohr, O. Graute, H. Hartenstein, and M. Zitterbart. Introducing probabilistic radio propagation models in omnet++ mobility framework and cross validation check with ns-2. In *Proc. of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops (Simutools '08), Marseille, France*, Simutools '08, pages 72:1–72:7, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[32] S. Liu, Y. Chen, W. Trappe, and L. J. Greenstein. Aldo: An anomaly detection framework for dynamic spectrum access networks. In *INFOCOM*, pages 675–683. IEEE, 2009.

[33] J. Meng, W. Yin, H. Li, E. Hossain, and Z. Han. Collaborative spectrum sensing from sparse observations in cognitive radio networks. *Selected Areas in Communications, IEEE Journal on*, 29(2):327–337, February 2011.

[34] A. Min, K. Shin, and X. Hu. Attack-tolerant distributed sensing for dynamic spectrum access networks. In *Proc. of the 17th IEEE International Conference on Network Protocols, 2009 (ICNP 2009), Princeton, NJ, USA*, pages 294 –303. IEEE, Oct. 2009.

[35] A. Min, K. Shin, and X. Hu. Secure cooperative sensing in ieee 802.22 wrans using shadow fading correlation. *Mobile Computing, IEEE Transactions on*, 10(10):1434 –1447, oct. 2011.

[36] S. Mishra, A. Sahai, and R. Brodersen. Cooperative sensing among cognitive radios. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 4, pages 1658–1663, June 2006.

[37] J. Mitola and J. Maguire, G.Q. Cognitive radio: making software radios more personal. *Personal Communications, IEEE*, 6(4):13–18, 1999.

[38] A. Mody, R. Reddy, T. Kiernan, and T. Brown. Security in cognitive radio networks: An example using the commercial ieee 802.22 standard. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1–7, Oct 2009.

[39] B. Naqvi, S. Murtaza, and B. Aslam. A mitigation strategy against malicious primary user emulation attack in cognitive radio networks. In *Emerging Technologies (ICET), 2014 International Conference on*, pages 112–117, Dec 2014.

[40] M. E. J. Newman. *Networks: An Introduction*. Oxford : Oxford University Press, 2011., 2011.

[41] N. Patwari and P. Agrawal. Effects of correlated shadowing: Connectivity, localization, and rf tomography. In *Proc. of the International Conference on Information Processing in Sensor Networks, 2008 (IPSN '08), St. Louis, Missouri, USA*, pages 82–93, 2008.

[42] F. Peng, H. Chen, and B. Chen. On energy detection for cooperative spectrum sensing. In *Information Sciences and Systems (CISS), 2012 46th Annual Conference on*, pages 1–6, March 2012.

[43] J. Redi and R. Ramanathan. The darpa wnan network architecture. In *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, pages 2258–2263, Nov 2011.

[44] M. Rehmani, A. Viana, H. Khalife, and S. Fdida. A cognitive radio based internet access framework for disaster response network deployment. In *Applied Sciences in Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on*, pages 1–5, Nov 2010.

[45] Y. Sagduyu. Securing cognitive radio networks with dynamic trust against spectrum sensing data falsification. In *Military Communications Conference (MILCOM), 2014 IEEE*, pages 235–241, Oct 2014.

[46] S. J. Shellhammer, S. S. N, R. Tandra, and J. Tomcik. Performance of power detector sensors of dtv signals in ieee 802.22 wrans. In *Proc. of the First International Workshop on Technology and Policy for Accessing Spectrum (TAPAS 2006), Boston, Massachusetts, USA*, TAPAS '06, New York, NY, USA, 2006. ACM.

[47] P. Steenkiste, D. Sicker, G. Minden, and D. Raychaudhuri. Future directions in cognitive radio network research. *NSF Workshop*, 2009.

[48] E. Tragos and V. Angelakis. Cognitive radio inspired m2m communications. In *Proc. of the 16th International Symposium of the Wireless Personal Multimedia Communications 2013 (WPMC '13), Atlantic City, New Jersey, USA*, pages 1–5. IEEE, June 2013.

[49] G. Trenkler. Statistical distributions: M. evans, n. hastings & b. peacock (1993): (2nd edition). new: John wiley. 170 pages, isbn 0-471-55951,[pound sign] 24.95. *Computational Statistics & Data Analysis*, 19(4):483–484, 1995.

[50] B. Wang and K. Liu. Advances in cognitive radio networks: A survey. *IEEE Journal of Selected Topics in Signal Processing*, 5(1):5 –23, Feb. 2011.

[51] K. Woyach and A. Sahai. The need for a new model of trust in spectrum and the case for spectrum jails. In *Proc. of the IEEE International Conference on Dynamic Spectrum Access Networks 2014 (DYSPAN 2014), McLean, Virginia, USA*, pages 427–438. IEEE, April 2014.

[52] X. Xing, T. Jing, W. Cheng, Y. Huo, and X. Cheng. Spectrum prediction in cognitive radio networks. *IEEE Wireless Communications*, 20(2):90–96, April 2013.

[53] X. Xing, T. Jing, Y. Huo, H. Li, and X. Cheng. Channel quality prediction based on bayesian inference in cognitive radio networks. In *IEEE INFOCOM*, pages 1465–1473, April 14-19 2013.

[54] K. Zeng, P. Paweczak, and D. Cabric. Reputation-based cooperative spectrum sensing with trusted nodes assistance. *Communications Letters, IEEE*, 14(3):226–228, March 2010.

[55] Y. Zhao, J. Reed, S. Mao, and K. K. Bae. Overhead analysis for radio environment mapenabled cognitive radio networks. In *Networking Technologies for Software*

Defined Radio Networks, 2006. SDR '06.1st IEEE Workshop on, pages 18–25, 2006.

APPENDIX - Rogue Signal Clustering Defense (RCD) Algorithm - the psuedo code that locates sensors affected by rogue signals in trust-based CSS protocols.

Algorithm 1 The RSF Cluster Detection Module

Function:**RCD**(A, S_{H_0}, S_{H_1})

Initialize cluster index $k \leftarrow 0$
Initialize set of protected sensors S_P
Initialize set of visited nodes V
Initialize Breadth-First-Search queue Q
Initialize set of clusters C_k
Initialize list clustering strength values Z_k
for all $s_i \in S_{H_1}$ **do**
  if $s_i \notin V$ **then**
    $k \leftarrow k + 1$
    add s_i onto C_k, V, and Q
    while Q is not empty **do**
      $s_q \leftarrow$ dequeue(Q)
      for all $s_j \in S_{H_1}$ **do**
        if $s_j \notin V$ **and** $A_{qj} = 1$ **then**
          add s_j onto C_k, V, and Q
        end if
      end for
    end while
    $\{d_i^{H_1}\}_k \leftarrow D(C_k)$
    $\{d_i^{\Delta}\}_k \leftarrow D'(C_k, S_{H_0})$
    $z_k \leftarrow Z(\{d_i^{H_1}\}_k, \{d_i^{\Delta}\}_k))$
    add z_k onto Z_k
  end if
end for
for all $z_k \in Z_K$ **do**
  if $|C_k| \geq C_{min}$ **and** $z_k > Z_\theta$ **then**
    $S_P \leftarrow S_P \cup C_k$
  end if
end for
return S_P

Vita

David Scott Jackson was born on July 5th of 1988 in Fort Benning, Georgia. He graduated from Brooke Point High School located in Stafford, Virginia in 2006. He received his B.S. and M.S. degrees in Computer Science from Virginia Commonwealth University in 2011 and 2013, respectively. Prior to joining Dr. Yu and Dr. Zang's security lab, he worked as an undergraduate research assistant in a VCU Biomedical Engineering Lab which resulted in his first accepted paper at the ASEE 2011 Conference, titled Development of Haptic Virtual Reality Gaming Environments for Teaching Nanotechnology. In April of 2013, he received the Outstanding Graduate Teacher Assistant Award by the VCU School of Engineering for that year. David is the recipient of numerous awards from intercollegiate computer science competitions, including "Most Innovative App" at Dominion Enterprise's Hackathon (HackU) in 2013, 2^{nd} place at UNC's Hackathon in 2014, and winner of the GhostRed CTF hosted by GE's cyber security division in 2014. As a Ph.D. candidate, he went on to publish one conference and two journal papers regarding his work in the area of wireless network security. In the summer of 2015, David joined the cyber-security firm, FireEye Inc., as a Sr. Software Engineer on the mobile security team.