ARTICLE

Mitigation of the Spectrum Sensing Data Falsifying Attack in Cognitive Radio Networks

Rajorshi Biswas[a], Jie Wu [a], Xiaojiang Du [a], and Yaling Yang [b]

[a] Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA;
[b] Bradley Dept. of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA.

ARTICLE HISTORY
Compiled July 22, 2020

ABSTRACT
Cognitive radio networks (CRNs), which offer novel network architecture for utilizing spectrums, have attracted significant attention in recent years. CRN users use spectrums opportunistically, which means they sense a channel, and if it is free, they start transmitting in that channel. If the status determination is wrong, secondary users (SUs) may unnecessarily interfere with the licensed primary user (PU). There are many spectrum sensing techniques, including cooperative spectrum sensing and database assisted spectrum sensing. In cooperative spectrum sensing, an SU makes a decision about the presence of the PU based on information from other SUs. The information used to make this decision follows different rules than other SUs' sensing information. Malicious SUs (MSUs) send false sensing information to other SUs so that they make wrong decisions about the spectrum status. As a result, an SU may transmit during the presence of the PU or may keep starving for the spectrum. Both consequences degrade the performance of CRNs. In this paper, we propose a reputation-based mechanism which can minimize the effects of MSUs on decision making in cooperative spectrum sensing. Some of the SUs are selected as distributed fusion centers (DFCs), that are responsible for making decisions about the presence of PU and informing the reporting SUs. A DFC uses weighted majority voting among the reporting SUs, where weights are normalized reputation. The DFC updates reputations of SUs based on confidence of an election. If the majority wins by a significant margin, the confidence of the election is high. In this case, SUs that belong to the majority gain high reputations. We conduct extensive simulations to validate our proposed model.

KEYWORDS
SSDF, spectrum sensing, cognitive radio networks, spectrum sensing data falsifying attack, security, spectrum security

1. Introduction

In a cognitive radio network (CRN), users can use a channel if it is not used by the licensed user. The licensed users are called primary users (PUs) and the CRN users are called secondary users (SUs). Detection of the PU transmission plays an important role in the throughput of a CRN. There can be two error cases: the PU is transmitting but an SU detects the channel to be free (false-negative), or the PU is not

CONTACT Rajorshi Biswas. Email: rajorshi@temple.edu

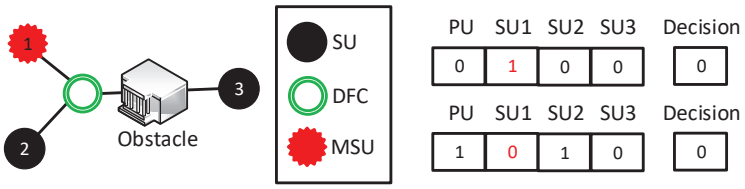


Figure 1.: Example of an SSDF attack.

transmitting but an SU detects the channel to be occupied (false-positive). A false-negative scenario leads the SU to transmit and cause interference with the PU, which is unexpected. A false-positive scenario prevents an SU from using the free channel, which reduces the CRN throughput. These kinds of detection errors are very common because of shadowing, multipath effects, path loss, and hidden terminals. SUs use the cooperative sensing mechanism to reduce the error rate [1]. In this mechanism, SUs share their sensing results with other SUs. An SU determines the channel status based on sensing information from itself and from others.

There are two kinds of architectures for cooperative spectrum sensing: distributed spectrum sensing and centralized spectrum sensing. In a distributed system, SUs broadcast their sensing information to their neighbors, and they make decisions according to "and", "or", "majority", or other rules. In a centralized system, all the SUs send sensing information to the fusion center (FC), and the FC makes the decision. SUs ask the FC for the channel status before starting transmissions. Cooperative sensing is also divided into two classes: soft combining and hard combining. In the soft combining method, SUs do not make decisions about the PU's presence; instead, they send their raw sensing information, including received energy and the signal-to-noise ratio, to the FC or to other SUs. The SUs or the FC make decisions based on raw information. In the hard combining method, every SU makes their own decision and shares that information to the FC or to other SUs. The information is basically an array of one bit information that represents the presence of PU in different time slots.

We consider that there are some malicious SUs (MSUs) in the system. The number of MSUs are less than the number of benign SUs. MSUs send incorrect sensing information to the FC or to other SUs to change the results. Some MSUs always say that the PU is absent, which leads to SU's transmission interfere with PU's transmission. Some MSUs always say that the PU is present, which prevents the SUs from using a free channel. Some MSUs are very smart and always say the opposite of the truth to fool the decision maker. We assume that the majority of SUs are benign so that one can argue that if the majority vote among the SUs will give the correct result. In the worse case, the MSUs and the benign SUs with wrong sensing information can win the vote. This type of attack is called spectrum sensing data falsifying (SSDF) attack.

In Fig. 1, SU 3 remains behind an obstacle. The error rate of SU 3 is greater than SU 2, but less than MSU 1 . We represent the PU presence as 1 and absence as 0. When the PU =0 (PU is absent), the sensing results of SUs 1, 2, and 3 are 1, 0, and 0, respectively. The majority decision is 0 (PU is absent), which is correct. When the PU=1 (PU is present), the sensing results of SUs 1, 2, and 3 are 0, 1, and 0, respectively. The majority decision is 0 (PU is absent), which is incorrect. Therefore, it is important to identify the MSUs and reduce their weights in election.

In this paper, we propose an algorithm to calculate the reputations of SUs. We introduce a system with distributed fusion centers (DFCs) that keep track of reputations

of the SUs. To the best of our knowledge, for the first time, we use an adaptive learning rate based on the confidence for reputation calculation. We consider the confidence of an election as the difference between the majority and minority population. When the confidence level is low, reputation increases or decreases at a lower rate. When the confidence level is high, reputation increases or decreases at a higher rate. None of the existing works use the confidence for reputation calculation, which is an important parameter. The adaptive learning rate helps the system identity the MSUs quickly and correctly. We conduct extensive simulations with multiple synthetic detests to compare our proposed cooperative spectrum sensing scheme with some existing schemes. Our main contributions are as follows:

- We propose an adaptive multiplicative reputation update method that uses confidence as a parameter.
- We propose a two level election scheme using a new combining method that also considers the confidence from the first level election.
- We conduct extensive simulations with different distributions and different settings to support our proposed model.

The remainder of the paper is as follows. Section 2 describes some related works. Section 3 describes the system and the attacker model. Our proposed SSDF mitigation scheme is presented in Section 4. Some existing and proposed reputation calculation schemes are presented in Section 5. Section 6 presents rules for combining other DFCs' decisions with other DFCs' observations. In Section 7, experimental results are presented. Finally, Section 8 concludes our work.

2. Related Work

There are many existing works on cooperative sensing under the SSDF attack. Solutions are based on the authentication of SUs [2], the clustering of benign SUs into a group, and the reputation-trust of SUs. Authors in [3,4] propose two different clustering algorithms based on the hamming distance among the sensing results of different timeslots of SUs. An associative rule mining based classification is proposed in [5]. Authors propose an apriori algorithm to get frequent subsets of the sensing results from all the SUs. The MSUs remain in the frequent subsets of the sensing results. Based on the probability of the PU's presence, SUs are classified into benign SUs and MSUs.

The reputation and trust based evaluation is well-studied in wireless sensor networks [6–8] for detecting malicious nodes. A trust-based spectrum sensing scheme against SSDF attack is proposed in [9]. In this method, the FC selects some of the SUs to make local decisions and the FC combines detection results based on their reliability. Authors in [10] propose a distributed spectrum sensing method in which reputations of SUs are computed based on deviation from the majority's decision. If an SU's sensing result is different from (or the same as) the majority, then its reputation is decreased (or increased) by one. A trust based dynamic collusive SSDF attack mittigation approach is proposed in [11]. The defense scheme called TFCA is based on trust fluctuation clustering analysis.

A PU emulation-based testing scheme, FastProbe, creates PU signals to test whether the SUs are reporting honestly or not [12]. They propose a scheduling algorithm to periodically test the SUs. This detection technique is now ineffective because there are a lot of mechanisms to detect PU emulation signals [13–17]. In addition, these

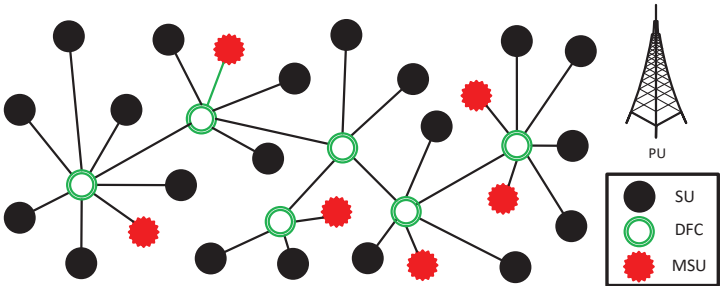


Figure 2.: Cooperative sensing system.

mechanisms are based on distribution, mean, and variance of energy and transmitter localization. An MSU can detect the PU emulated signal and report the correct result in that timeslot to get a high reputation. Then, it can keep reporting false results in other timeslots. The reputation-trust based systems use history to calculate SUs' reputation. On every observation, their reputation is updated based on some rules. None of them use the confidence of an election for updating reputation.

The SSDF attack can be launched with spoofing and jamming attack [18] using a strategy to maximum attacking strategy using spoofing and jamming.The attacker utilizes an optimal power distribution to maximize the attack effects. Spoofing and jamming attacks are launched dynamically to interfere with the maximum number of signal channels. However, we are not considering a mixed type of attack in this research.

3. System and Attacker Model

In this section, we define the attacker and the system model. For ease of referencing we list all the notations in Table 1.

3.1. System Model

We consider a CRN with some SUs and a PU. The PU frequently goes on and off in its licensed channel and the PU presence is uniformly random. All the SUs are located in a small area and impacted by a PU. In addition, the local sensing results of SUs are mutually exclusive. The SUs sense the PU's licensed channel and periodically send sensing results directly to the DFC. DFCs are also SUs, but instead of only sensing, they work as aggregators of others' sensing results. DFCs make decisions based on the sensing results sent from their neighbors.

An SU can become a DFC if it meets certain criteria. A connected dominating set (CDS) is formed among the SUs. The CDS is used in wireless sensor networks (WSN) to select relay nodes to broadcast a message. The main benefit of using CDS in a WSN is that every node can reach a relay node within an 1-hop neighborhood. The difference between CRN and WSN is that nodes are static in WSN but mobile in a CRN. Therefore, computing CDS in a CRN is similar to mobile ad hoc networks. There are some existing algorithms for constructing a CDS. Authors in [19] present a node-degree-based dominating node selection process. In that process, all nodes are initially colored white. Then, the node with the maximum node degree is selected as the root

and colored black. All the neighbors of the root are colored gray. The process proceeds by selecting a gray node who has maximum white neighbors. The node is colored black and its white neighbors are colored gray. The process is complete if there is no more white nodes. Finally the black nodes form the CDS. A minimum spanning tree-based CDS is proposed in [20]. First, a spanning tree is constructed. Then, nodes on the tree are colored white and labeled according to topological order. Nodes are marked based on their positions starting from the root. When a node is marked black, its subsequent node is labeled black if it does not have a black neighbor. Each black node except the root node selects a neighbor with the largest label that is still smaller than its own label and marks it gray. Then, the black nodes and gray nodes become the CDS. A marking process-based CDS construction is proposed in [21]. A node is marked true if it has two unconnected neighbors and the set of marked nodes forms a CDS. This marking process generally produces a CDS with many nodes. Consequently, the CDS needs to cut off some of the nodes. If a node's neighbors are covered by other nodes, then it can be removed. An energy change-based CDS formation is proposed in [22]. The reputation of an SU can be used as weight in this scheme which will prevent the MSUs from becoming a DFC. When two nodes come closer, the energy of the "hello" message increases, and when they move away from each other, the energy decreases. Based on the increment or decrement, the weight of each edge is updated. An edge between two nodes means that the two nodes can communicate with each other. A positive weight value on an edge indicates that the corresponding nodes are coming close to each other. A node's weight is the summation of the weights of its edges. A node with a higher weight is selected as a dominating node by lower-weighted neighboring nodes. This selection process finally forms a CDS in the network. A weighted backbone using a small communication cost is proposed in [23]. This method also uses a CDS formation which works for both homogeneous and heterogeneous networks. The main advantage of having a reputation/ weight-based CDS formation method is that the MSUs cannot get selected as CDS. In this paper, we do not focus on the selection of DFC.

In Fig. 2, the green SUs are selected as CDS. The black SUs are not members of CDS and they can find a green SU within one hop. The SUs in CDS become DFCs. Every SU sends their 1 bit sensing result to the neighboring DFC. The DFC runs a weighted majority voting among the received sensing results and updates reputation values of the SUs. Then the DFC shares its result with other DFCs. After receiving results from other DFCs, a DFC makes a final decision combining its own and others' voting results.

3.2. Attacker Model

The DFCs know only the SUs who report sensing information to them. A DFC or FC does not know how many SUs are benign or malicious. We assume that the number of MSUs is smaller than the number of benign SUs. We assume that all the SUs are located in a small area and impacted by a PU and the local sensing result of SUs are mutually exclusive. The misdetection probability and the false alarm probability are similar for all benign SUs. On the other hand, the detection probability and the false alarm probability are higher for MSUs. Based on the attacking behavior, we can classify the attacker's strategy in four classes:

(1) "Random Yes" Attack: The MSU sends "1" to the DFC regardless of the sensing result with α probability. This may happen because the MSU does not have a

Table 1.: Table of notations

M	Distributed fusion center
I	Number of SUs report to M
su_i	i th SU.
PU	1 (or 0) for PU present (or absent).
$w_t[i]$	Weight of su_i at timeslot t
$D_t[M]$	Decision of M at timeslot t
$x_t[i]$	Sensing result of su_i at timeslot t
$r_t[i]$	Reputation of su_i at timeslot t
$\rho_t[M]$	Confidence of DFC M at timeslot t
δ	Sliding window size
η	Learning rate
$D'_t[M]$	Decision of M from neighbor DFCs' decisions
$\rho'_t[M]$	Confidence of DFC M from neighbor DFCs' decisions
$D''_t[M]$	Final decision of M
$\rho''_t[M]$	Final confidence of M
Th	Threshold for first level election
Th'	Threshold for second level election
Th''	Threshold for combining election results
μ	Weight of first level election

sensing module and tries to fool other SUs. This type of MSU succeeds when it makes DFC's decision "1" when there is no PU present. When $\alpha = 1$, the MSU always sends "1" to the DFC, which is called the "Always Yes" attack. To maximize success, the MSU tries to send as many false reports as possible.

(2) "Random No" Attack: This attack is the opposite of the "Random Yes" attack. The MSU sends "0" to the DFC regardless of the sensing result with α probability. This type of MSU succeeds when it makes the DFC's decision "0" when PU is present. When $\alpha = 1$, the MSU always sends "0" to the DFC; this is called the "Always No" attack. The consequence of this attack is more devastating than the "Random Yes" attack. SUs that know the PU is absent start transmitting, resulting in interference with the PU.

(3) "Random False" Attack: The MSU sends the opposite sensing result to the DFC with a probability of α. That means when the MSU's sensing result is "1", it sends "0", and when it is "1", it sends "0" to the DFC. When $\alpha = 1$, the MSU always sends the opposite sensing result. This kind of attack is basically a mixture of "Random Yes" and "Random No" attacks.

(4) "Completely Random" Attack: The MSU sends random sensing information with a probability of α. MSU selects "0" or "1" randomly and overwrites the result in a time slot with α probability. $\alpha = 1$ means the MSU sends a random sensing result in each time slot. This type of MSU may not be intentionally malicious, but it is a threat to the system. Unintentionally, MSUs try to hide their sensing process failures by sending random sensing information in the time slot.

Figure 3 depicts these attacks when $\alpha = 1$. It shows the sensing result of a benign SU, an "Always No" MSU, and of an "Always Yes" MSU for timeslot t_o to t_7.

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
Benign SU	0	1	1	0	0	1	0	1
"always no"	0	0	0	0	0	0	0	0
"always yes"	1	1	1	1	1	1	1	1
"always false"	1	0	0	1	1	0	1	0
"completely random"	1	1	0	0	1	1	1	0

Figure 3.: Different attacking strategy of MSU.

4. Proposed Cooperative Sensing Architecture

Online machine learning is referred to as a learning system where data is available to the system in a sequential manner. In our system, SUs keep sending sensing results of each timeslot to DFCs. Data from nearby SUs go to the DFC in a sequential manner. Let us consider that SU M becomes a DFC and I SUs report to M. At time t, the sensing result from $su_1, su_2, .., su_I$ goes to M. In addition, M keeps the weight and reputation of each neighboring SU. When the sensing results from neighboring SUs arrive at M, it calculates the sensing result based on the weighted votes of the SUs' results.

$$D_t[M] = \begin{cases} 1, & for \sum_{i=0}^{I} w_t[i] D_t[i] \geq Th \\ 0, & otherwise \end{cases} \tag{1}$$

Here, Th is a threshold, which determines the portion required to win the vote. For example, if the weights of all SUs are equal, then $Th = 0.5$ means the "majority" voting, $Th = 1$ means the AND voting, and $Th = 0$ means the OR voting. We express the reputation $r_t[i]$ of su_i at time t as following:

$$\begin{aligned} x_t[i] &= \begin{cases} 1, & \text{if } D_t[i] = D_t[M] \\ -1, & \text{if } D_t[i] \neq D_t[M] \end{cases} \\ r_t[i] &= f(*) \end{aligned} \tag{2}$$

Here, $D_t[i]$ and $D_t[M]$ denote the one bit decisions about the PU presence of su_i and M at time t. $\rho_t[M]$ denotes the confidence of election at the DFC M at time t. $f(*)$ is called the weight update function (WUF). Different WUFs take different parameters, including a common parameter x_t. We are not defining the parameters to make WUFs general. Examples of different WUFs will be discussed in Section 5. Let C_0 of the SUs report that the PU is absent and C_1 of them report that the PU is present ($C_0 + C_1 = I$) to M at timeslot t. So, the confidence level of M at time t is:

$$\rho_t[M] = \left| \frac{C_0 - C_1}{C_0 + C_1} \right| \tag{3}$$

When someone wins by a significant difference in vote, we conclude that the confi-

Algorithm 1 Online Learning-Based Spectrum Sensing

Input: D of DFC M, where $D_t[i]$ is sensing result of neighboring SU i at time t.
Output: Weights for voting w.

procedure CONTINUOUS-UPDATE(D)
  Initialize reputation $r_0[i]$ for SU i for $i \in$ NEIGHBOR(M) and $t \leftarrow 0$.
  while true **do**
    $C_0 \leftarrow$ number of 0 s in D_t, $C_1 \leftarrow$ number of 1 s in D_t.
    if $\sum_{i=0}^{I} w_i D_t[i] > Th$ **then**
      $D_t[M] \leftarrow 1$.
    else
      $D_t[M] \leftarrow 0$.
    $\rho_t \leftarrow \left|\frac{C_0 - C_1}{C_0 + C_1}\right|$
    $x_t[i] = \begin{cases} -1, & \text{if } D_t[i] \neq D_t[M] \\ 1, & \text{if } D_t[i] = D_t[M] \end{cases}$
    $r_{t+1}[i] = f(*)$
    $w_{t+1}[i] = \frac{r_{t+1}[i]}{\sum_{i=0}^{I} r_{t+1}[i]}$
    Output w_{t+1}.
    $t \leftarrow t + 1$.

dence of the election is high. If the confidence is high, then the effect of the result will also be high. That is why we use the proposed adaptive multiplicative WUF in the Algorithm 1. The complete process is shown in Algorithm 1.

The stated problem is similar to the experts' opinion aggregation problem where an aggregator with little knowledge tries to come to a Yes/No decision. Before making any decision, the aggregator asks all of its nearby experts for their opinions. Experts respond with Yes/No answers. Based on their decisions, the aggregator makes its own decision and calculates their reputation values. Reputation values are used for future decision making; a high reputation value means that the experts' decisions will have priority over others with low reputation values. Some literature assume that the aggregator knows the ground truth of the result in the next timeslot. They can update the reputation values of experts based on differences between their answers and the ground truth. Our problem becomes more challenging because there is no ground truth. The most challenging part of the problem is to find an SUitable WUF. We discuss some WUFs in the next section.

5. Different WUFs

In this section, we present the existing linear, and multiplicative WUFs with or without a sliding window. We propose our adaptive-multiplicative WUF with a sliding window.

5.1. Linear WUF

Some articles like [10,24] use linear WUF to update the reputations of sensor nodes in WSN. At $t = 0$, an aggregator can assume all the SUs are benign (highly reputed) and decrease reputation based on their behavior. The drawbacks of this assumption are that the system needs some initial time to set up and an MSU can start again

with a new ID when its reputation becomes too low. Let us assume that at $t = 0$, an aggregator assumes all the SUs' reputations are 0. After evaluation, the aggregator increases the SU's reputation. Reputation update depends on two types of information: first-hand information and second-hand information. First-hand information refers to an SU's own observed information. Second-hand information refers to the reputations of other SUs. Based on the first-hand information, the reputations update is done as following:

$$f(*) = f(r_t[i], x_t[i]) = \mu \times r_t[i] + (1 - \mu) \times x_t[i] \tag{4}$$

Here, μ is between [0, 1] and it determines how much the current observation affects the reputation. If the SU's prediction is wrong, then the last part of the equation $((1 - \mu) \times x_t[i])$ is negative and the reputation is reduced.

5.2. Multiplicative WUF

In multiplicative WUF, reputations are increased or decreased by a factor. $f(*)$ for the multiplicative WUF can be defined as following:

$$f(*) = f(r_t[i], x_t[i]) = r_t[i] \times exp(\eta x_t[i]) \tag{5}$$

Here, η is the learning rate, and it determines the portion of contribution from the current observation to the reputation. Another version of this multiplicative WUF considering the sliding window can be expressed as the following:

$$f(*) = f(r_t[i], x_t[i]) = r_t[i] \times \frac{exp(\eta x_t[i])}{exp(\eta x_{t-\delta}[i])} = r_i \times exp(\eta(x_t[i] - x_{t-\delta}[i])) \tag{6}$$

Here δ is the effective evaluation period of the SUs. Dividing it by a factor $exp(\eta x_{t-\delta}[i])$ nullifies the reputation contribution at timeslot $t - \delta$. Therefore, DFCs need to store δ number of past sensing results for every reporting SU. DFCs do not need to store past confidence because they can recalculate it from the sensing result.

5.3. Adaptive Multiplicative WUF

In our society, the reputations of people do not rise or sink linearly. People have to work hard to become popular in politics, school, or work. Once someone becomes popular, his/her small positive activities raise his/her popularity to a great extent. Our reputation calculation scheme is motivated by this social fact. The higher an SU's reputation is, the more it can be increased (or decreased) by correct (or wrong) sensing. If a large number of SUs agree with the DFC's result, then its confidence level is higher. On the other hand, if almost half of the SUs disagree with the DFC's result, then its confidence level is lower. Therefore, we propose adaptive multiplicative WUF which is slightly different than the multiplicative WUF. Instead of using a constant learning rate η, we multiply it by the confidence of the election. The WUF can be expressed as following:

$$f(*) = f(r_t[i], x_t[i]) = r_i \times exp(\eta(\rho_t[M]x_t[i] - \rho_{t-\delta}[M]x_{t-\delta}[i])) \tag{7}$$

Table 2.: Job Interview Scoring Summary

Student		Interviewer	
Confidence	Answer	Confidence	Score
High	Correct	High	High
High	Wrong	High	-High
Low	Correct	High	Low
Low	Wrong	High	-Low
High	Correct	Low	Low
High	Wrong	Low	-Low
Low	Correct	Low	Lower
Low	Wrong	Low	-Lower

6. Hard but Soft Combining Rule

Consider a scenario where a student is answering tough questions in a job interview. Some of the answers are known and some are unknown to the student. The student sometimes answers with high confidence and sometimes with low confidence. Some of the questions are ambiguous and even the interviewer is confused about the answer. In this scenario, the interviewer follows some simple rules. When the student is very confident and his answer is correct, he gets a high score when interviewers confidence is high. When the student answers with less confidence and his answer is correct, he gets a low score when interviewers confidence is high. Table 2 summarizes this concept.

Observing Table 2, we see that the confidence of interviewer and student are multiplied to get the score. The correctness of the answer determines the sign of the score. Based on this principle, we propose the "Hard but Soft" combining rule. In soft combining methods, SUs send their raw sensing information to the FC. In hard combining rules, SUs send one bit information to the FC whether or not the PU is present to the FC. Our proposed "Hard but Soft" rule is in between. This combining rule is applicable when DFCs share their results with other DFCs. Each DFC result has a confidence level. Each DFC shares their one bit result and the confidence level of the result with other neighboring DFCs. When a DFC's result matches (or does not match) with the majority's result and both its confidence level and the aggregator DFC's confidence level is high, then its reputation increases (or decreases) significantly. When a DFC's confidence or the aggregator DFC's confidence is low, the reputation of the DFC increases/decreases at a low rate. When both of the DFCs' confidences are low, the DFC's reputation increases/decreases at a lower rate.

Let P and Q be two neighboring DFCs. P receives a result about the PU's presence $D_t[Q]$ and confidence of result $\rho_t[Q]$ at time t. P determines its decision $D_t[P]$ and confidence $\rho_t[P]$ using weighted majority rules. Then it compares that decision with the decision from the DFC Q at the timeslot t. The reputation of Q is updated according to the following:

$$r_t[Q] = \begin{cases} r_{t-1}[Q] \times exp(\eta \rho_t[P] \rho_t[Q]), & D_t[P] = D_t[Q] \\ r_{t-1}[Q] \times exp(-\eta \rho_t[P] \rho_t[Q]), & \text{otherwise} \end{cases} \tag{8}$$

The difference between Equation 5 and Equation 8 is, Equation 5 uses only the confidence of the aggregator and Equation 8 uses both the aggregator's and sender's

confidence level. This reputation update scheme increases truthfulness. When a DFC lies with high confidence, then it has a high chance of being caught and penalized by the aggregator. On the other hand, lying with low confidence will not affect the aggregator DFC's result significantly. Let there be N neighboring DFCs of the DFC M who send their aggregated results with confidence levels. So, M's result from other DFCs is as follows:

$$\begin{aligned} D'_t[M] &= \begin{cases} 1, \sum_{i=0}^{N} w_t[i]\rho_t[i]D_t[i] > Th' \\ 0, otherwise \end{cases} \\ \rho'[M]_t &= \frac{1}{N}\sum_{i=0}^{N}\rho_t[i] \end{aligned} \tag{9}$$

Here, $w_t[i]$ is the weight (normalized reputation) of the DFC i at time t. $\rho_t[i]$ is the confidence of the result of $D_t[i]$ of DFC i. ρ'_t is the confidence of the result $D'_t[M]$ from second hand information, and Th' is another system variable that is similar to Th, which determines the portion required to win the vote.

6.1. Combining Results from SUs and other DFCs

As discussed in the previous sections, DFCs get information from two types of sources: SUs and other DFCs. Two types of sources produce two results which may or may not be the same. The final result is a combination of information from SUs and other DFCs. Let's assume that DFC M's calculated result from first hand information (neighboring SUs sensing information) is $D_t[M]$, and the confidence of the result is $\rho_t[M]$. From the second hand information (other DFCs shared results) M's decision is $D'_t[M]$ and the confidence of the result is $\rho'_t[M]$. The final result is $D''_t[M]$ and confidence of the final result is $\rho''_t[M]$.

$$\begin{aligned} D''_t[M] &= \begin{cases} 1, \, D_t[M]\rho_t[M]\mu' + D'_t[M]\rho'_t[M](1-\mu') > Th'' \\ 0, \, otherwise \end{cases} \\ \rho''_t[M] &= \rho_t[M]\mu' + \rho'_t[M](1-\mu') \end{aligned} \tag{10}$$

Here, μ' is a system variable, which determines how much a DFC will believe its neighboring SUs. The value of μ' can change for different DFCs. For example, if a DFC finds that its reputation among other DFCs is very low, then it can reduce its μ' to give less emphasis to its neighboring SUs. This adjustment is helpful when a DFC is surrounded by many MSUs. Th'' is another system variable similar to Th and Th'.

6.2. Performance Analysis

In the worst case, we assume the population of the benign SU is N and the population of the MSU is $N-1$. MSUs use the "always opposite" attack strategy. The accuracy (probability of correct sensing) of benign SU is p and MSUs' accuracy is very low ($\approx 0\%$). Then, the probability distribution of number of correct sensing x, is denoted by $P(x)$.

Table 3.: Dataset description

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
# of SUs	10	10	10	50
# of MSUs	10	10	7	40
SU error rate	[0, 0.3]	[0, 0.3]	[0, 0.3]	[0, 0.3]
MSU error rate	[0.8, 1]	[0.8, 1]	[0.7, 1]	[0.7, 1]
# of time slots	200,000	200,000	500	3000

$$P(x) = \binom{N}{x}(1-p)^x(p)^{N-x} \tag{11}$$

At the beginning, the weights/reputation of all SUs are equal. Therefore, only at $x = N$, the DFC can produce the correct sensing result. The expected increase in the reputation of a benign SU is by a factor of $P(N)exp(\eta\rho[M])$. On the other hand, MSU's reputation will decrease by a factor of $P(N)exp(\eta\rho[M])$. For $x < N$, MSUs are the majority and the DFC's result would be wrong. Therefore, an MSU's expected increase in reputation is $\sum_{x=0}^{N-1} P(x)exp(\eta\rho[M])$. In order to produce the correct result, the DFC should set parameters so that

$$\sum_{x=0}^{N-1} P(x)exp(\eta\rho[M]) < P(N)exp(\eta\rho[M]) \tag{12}$$

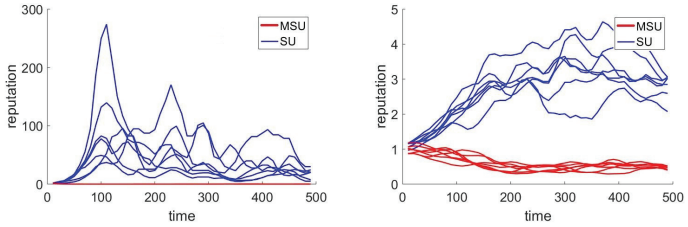
We compare the tolerance limit of our proposed WUF and the exponential WUF of 99 SUs (50 benign SUs and 49 MSUs). We define the tolerance limit as the error rate which violates the Equation 12. The tolerance limit of the proposed WUF is greater than the multiplicative WUF. The tolerance limit remains constant for the multiplicative WUF.

7. Experiments and Simulations

In this section, we present the experimental settings and simulation results conducted to support our proposed model.

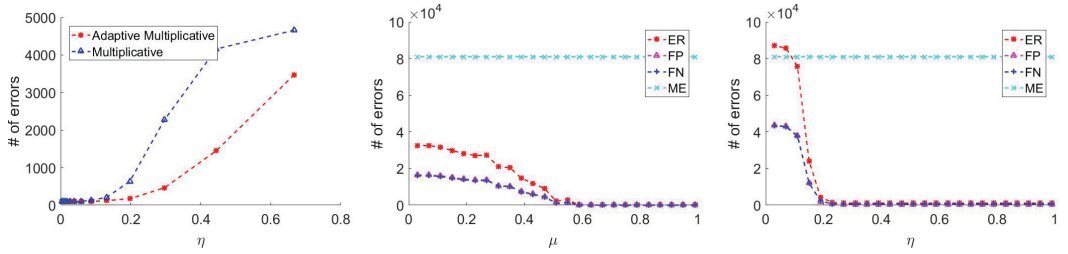
7.1. Comparison among different WUFs

We compare the performances of linear, multiplicative, and adaptive multiplicative WUFs for two datasets. We consider 10 benign SUs with sensing error rates within $[0, 0.3]$ and 10 MSUs with error rates within $[0.8, 1]$. Both datasets have sensing results over $200,000$ timeslots. In dataset 1, the MSUs show their malicious behavior from the beginning. In the dataset 2, the MSUs show benign behavior to build their reputations up to $100,000$ timeslots. From the $100,001$th timeslot, the MSUs start sending the wrong sensing results. We also have a couple of small datasets. Dataset 3 contains 10 SUs and 7 MSUs of 3000 timeslots and dataset 4 contains 50 SUs and 40 MSUs of 3000 timeslots. MSUs and SUs in datasets 3 and 4 have error rates within $[0.7, 1]$ and $[0, 0.3]$, respectively. Table 3 presents the detailed information about the datasets.

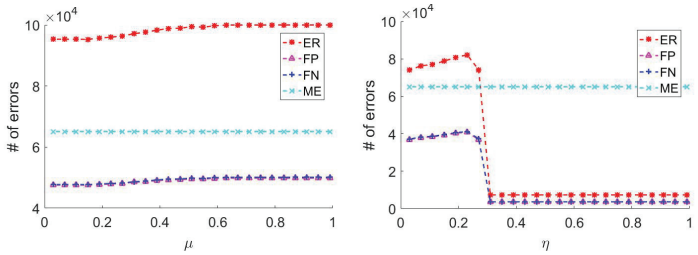


(a) Multiplicative WUF. (b) Adaptive multiplicative WUF.

Figure 4.: Comparison of reputations for dataset 3.



(a) Adaptive-multiplicative and multiplicative WUFs. (b) Linear WUF for dataset 1. (c) Adaptive-multiplicative WUF for dataset 1.



(d) Linear WUF for dataset 2. (e) Adaptive-multiplicative WUF for dataset 2.

Figure 5.: Simulation results.

Fig. 4 shows reputations of MSUs and SUs for 500 timeslots in dataset 3. We plot the reputation of multiplicative WUF in Fig. 4(a). We can observe that the reputations of SUs are very high compared to the MSUs. Reputations of SUs and MSUs are close to 0 and varied from from 1 to 290. Because of the high variation in reputations of SUs, any mistake in sensing results by a highly reputed SU results in changing the election result. We plot the reputation of adaptive multiplicative WUF in Fig. 4(b). The reputations of SUs are not as high as that of the multiplicative WUF. The variation in reputations of SUs or MSUs are not as high as the variation in reputations in multiplicative WUF. As a result, all the SUs (or MSUs) get similar priority in election. Therefore, any mistake by an SU cannot affect the election result largely.

Fig. 5(a) shows a comparison between adaptive multiplicative and multiplicative WUFs for dataset 1. It is observed that the number of errors in adaptive multiplicative WUF is less than that of the multiplicative WUF. Figs. 5(b) and 5(c) show the number of total errors (ER), false positive (FP), false negative (FN), and error in simple majority voting (ME) in linear and adaptive multiplicative WUFs for different values of μ and η in dataset 1. From the plots, we can observe that the number of errors in the linear weight update is very small (0.02%) when μ is within [0.6, 1.0]. On the other hand, the number of errors in the adaptive multiplicative WUF is higher than

Table 4.: FC and DFC comparison

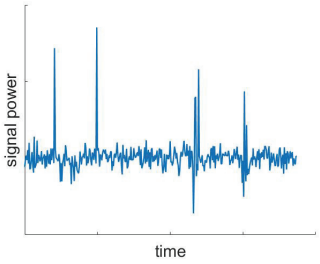
	Fig. 6(b)	Fig. 6(c)
SU error rate	[0,0.1]	[0,0.2]
MSU error rate	[0.7,1]	[0.8,1]
Lowest affected SU [FC,DFC]	[0%, 0%]	[0%,0%]

the linear WUF (0.6%). When η is within [0.2, 1.0], the linear WUF does better than multiplicative in the first scenario. However, MSUs can be as clever as they are in the second scenario, where they hide their original behavior until they build good reputations. In that scenario, the multiplicative WUF works better. Figs. 5(d) and 5(e) show the ER, FP, FN, and ME in linear and adaptive multiplicative WUFs for different values of μ and η in dataset 2. The lowest error rate with the linear update function is 49%. On the other hand, the multiplicative WUF error rate is almost stable at 3.6% for a learning rate within [0.3, 1]. The simple majority rule shows a 40% error for the dataset 1 and a 32% error for the dataset 2. We also see that the false positive and false negative rates are almost same, because we assume the PU's presence is uniformly random. Therefore, we conclude that the adaptive multiplicative WUF performs better than linear WUF and simple majority rules.

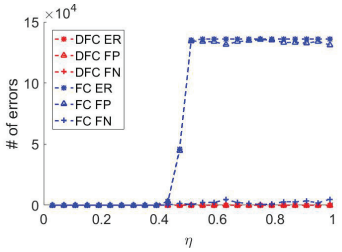
7.2. Simulation with real primary user Data

In the experiments above, we consider the PU's presence in each timeslot to be random. To get the PU's behavior, we observe a 2.4 GHz Wi-Fi band and assume Wi-Fi users to be the PU. We capture the signal power of the 6th channel (2.437Ghz 20Mhz channel) of the 2.4GHz band and use that information for the PU emulation. We observe that the PU remains OFF for long stretches of time before suddenly coming ON at some timeslots. Fig. 6(a) shows the PU behavior for certain time. We experiment with $1,000$ CR users where 50% of them are MSUs. We generate sensing results from $1,000$ users according to their sensing error rates, and we compare the conventional FC-based architecture with our DFC-based architecture in terms of the number of users affected when a wrong decision is made. In the conventional FC-based architecture, all sensing results are sent directly to the FC so that every SU is affected by the decision made by the FC. On the other hand, only the SUs that report to the DFC are affected by a wrong decision.

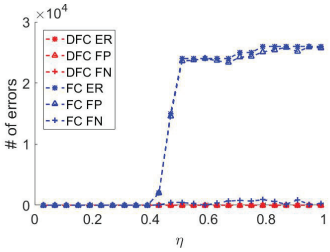
Figs. 6(b) and 6(c) show the number of error affected users (ER), false positive affected users (FP), and false negative affected users (FN) for $2,000$ timeslots for different learning rates (η) of both the FC and DFC based architectures. We keep $\mu' = 0.5$ for these experiments. We observe that false positive affected users are about 96% of the total affected users due to the fact that the PU remains off in most timeslots. For both systems, we use adaptive multiplicative WUFs. From Figs. 6(b) and 6(c), we see that for $\eta < 0.4$, both systems have no affected users. One can argue that if we set a low ($\eta < 0.4$) learning rate, we do not need the DFC-based system. The low learning rate is dangerous in a scenario where SUs/MSUs frequently changes behavior. For example, in Fig. 5(e), the low learning rate ($\eta < 0.3$) shows a large number of errors. From this perspective, the DFC-based architecture shows higher robustness. For η within $[0, 1]$, the DFC-based system shows no error. From the figure, we can conclude that the DFC-based system works better than the conventional FC-based system. Table 4 shows detailed parameters of the simulations.



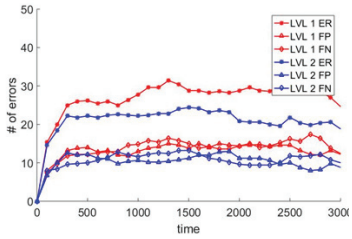
(a) Received signal power.



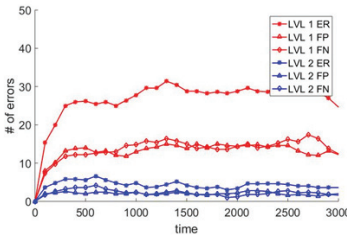
(b) FC and DFC based system for dataset 1.



(c) FC and DFC based system for dataset 2.



(d) Uniform MSU distribution for dataset 4.



(e) Clustered MSU distribution for dataset 4.

Figure 6.: More simulation results.

Figs. 6(d) and 6(e) show the number of errors (ER) of FC-based and DFC-based architectures for $3,000$ timeslots for different MSUs distributions. We use dataset 4 and keep $\eta = 0.01$ for this experiment. In the uniform distribution, all the MSUs are distributed uniformly. On the other hand, in the clustered distribution, 20% MSUs are distributed uniformly and the remain 80% are distributed in a clustered way (existence of a neighboring MSU increases probability of being an MSU). In the uniform distribution, we observe that the FC-based and DFC-based architecture show that over a period of 100 timeslot the number of errors are approximately 30 and 22. False positive and false negative rates are similar for all timeslots. In the clustered distribution, the FC-based and DFC-based architecture shows the number of errors are approximately 30 and 4. False positive and false negative rates are also similar. The number of errors remains the same in clustered FC-based architecture. This is because, the distribution of MSUs does not matter to FC as the number of MSUs and SUs are the same. The number of errors is reduced in clustered in DFC-based architecture. This is because the clustered MSUs can win at some of the DFCs in the first level election, but in the second level election, they fail.

Therefore, from the experiments we can conclude that our DFC-based architecture with adaptive multiplicative WUF is more robust to parameter settings. The number of errors and affected SUs are also less than other approaches mentioned above.

8. Conclusion

Though cooperative spectrum sensing with the FC shows great performance when detecting the PU's presence in CR networks, it suffers from SSDF attacks. We propose a CDS-based distributed spectrum sensing mechanism where some SUs become DFCs. The DFCs collaborate on spectrum sensing information sent by SUs. We propose an adaptive multiplicative WUF for reputation updates of SUs, which shows a better

performance compared to conventional multiplicative and linear WUFs. We consider a 2.4Ghz Wi-Fi channel as an unlicensed channel and Wi-Fi users as PUs, which is more realistic than assuming a random PU's presence. We also show that the DFC-based system performs consistently and better than the conventional FC-based system.

Acknowledgments

This research was supported in part by NSF grants CNS 1757533, CNS1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS1460971, and IIP 1439672.

References

[1] Y. Dai and J. Wu, "Cooperation Scheme For Distributed Spectrum Sensing In Cognitive Radio Networks," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 1, no. 4, 9 2014.

[2] J. Dai, J. Liu, C. Pan, J. Wang, C. Cheng, and Z. Huang, "MAC Based Energy Efficiency in Cooperative Cognitive Radio Network in the Presence of Malicious Users," *IEEE Access*, vol. 6, 2018.

[3] S. Nath, N. Marchang, and A. Taggu, "Mitigating SSDF attack using K-medoids clustering in Cognitive Radio Networks," in *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct 2015.

[4] K. Rina, S. Nath, N. Marchang, and A. Taggu, "Can clustering be used to detect intrusion during spectrum sensing in cognitive radio networks?" *IEEE Systems Journal*, vol. PP, no. 99, 2017.

[5] S. Bhattacharjee, R. Keitangnao, and N. Marchang, "Association rule mining for detection of colluding SSDF attack in Cognitive Radio Networks," in *2016 International Conference on Computer Communication and Informatics*, Jan 2016.

[6] T. Anantvalee and J. Wu, "Reputation-Based System for Encouraging the Cooperation of Nodes in Mobile Ad Hoc Networks," in *2007 IEEE International Conference on Communications*, Jun 2007.

[7] A. Srinivasan, J. Teitelbaum, and J. Wu, "DRBTS: Distributed Reputation-based Beacon Trust System," in *Proceedings of the 2Nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, 2006.

[8] X. Du and H.-H. Chen, "Security in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, 2008.

[9] F. Zeng, J. Li, J. Xu, and J. Zhong, "A Trust-Based Cooperative Spectrum Sensing Scheme against SSDF Attack in CRNs," in *2016 IEEE Trustcom/BigDataSE/ISPA*, Aug 2016.

[10] R. Chen, J. M. Park, and K. Bian, "Robust Distributed Spectrum Sensing in Cognitive Radio Networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, Apr 2008.

[11] F. Zhao, S. Li, and J. Feng, "Securing cooperative spectrum sensing against DC-SSDF attack using trust fluctuation clustering analysis in cognitive radio networks," *Wireless Communications and Mobile Computing*, 2019.

[12] T. Bansal, B. Chen, and P. Sinha, "FastProbe: Malicious user detection in Cognitive Radio Networks through active transmissions," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, Apr 2014.

[13] S. Anand, Z. Jin, and K. P. Subbalakshmi, "An Analytical Model for Primary User Emulation Attacks in Cognitive Radio Networks," in *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Oct 2008.

[14] D. Salam, A. Taggu, and N. Marchang, "An effective emitter-source localisation-based PUEA detection mechanism in cognitive radio networks," in *2016 International Conference on Advances in Computing, Communications and Informatics*, Sep 2016.

[15] Y. Liu, P. Ning, and H. Dai, "Authenticating Primary Users' Signals in Cognitive Radio Networks via Integrated Cryptographic and Wireless Link Signatures," in *2010 IEEE Symposium on Security and Privacy*, May 2010.

[16] C. Chen, H. Cheng, and Y. D. Yao, "Cooperative Spectrum Sensing in Cognitive Radio Networks in the Presence of the Primary User Emulation Attack," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, Jul 2011.

[17] R. Chen, J. M. Park, and J. H. Reed, "Defense against Primary User Emulation Attacks in Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, Jan 2008.

[18] K. Gai, M. Qiu, Z. Ming, H. Zhao, and L. Qiu, "Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2431–2439, 2017.

[19] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Communications, 1997. ICC '97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on*, vol. 1, Jun 1997.

[20] K. M. Alzoubi, P. J. Wan, and O. Frieder, "Distributed heuristics for connected dominating sets in wireless ad hoc networks," *Journal of Communications and Networks*, vol. 4, no. 1, Mar 2002.

[21] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," in *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999.

[22] S. Leu and R.-S. Chang, "A weight-value algorithm for finding connected dominating sets in a MANET," *Journal of Network and Computer Applications*, vol. 35, no. 5, 2012.

[23] Yu Wang, Weizhao Wang, and Xiang-Yang Li, "Efficient distributed low-cost backbone formation for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 7, 2006.

[24] S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, 2002.