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

Rajorshi Biswas, Jie Wu, and Xiaojiang Du
Department of Computer and Information Sciences
Temple University, Philadelphia, PA, USA
{rajorshi, jiewu, dux}@temple.edu

***Abstract*—Cognitive radio networks (CRNs), offering novel network architecture for utilizing spectrum, have attracted significant attention in recent years. In CRNs, secondary users (SUs) first determine the status of a channel; if it is free, they start transmitting. If the status determination is wrong, SUs may unnecessarily interfere with the licensed primary user (PU). In cooperative spectrum sensing, a SU makes a decision about the presence of the PU based on its own and other SUs' sensing results. Malicious SUs (MSUs) send false sensing results to SUs so that they make wrong decisions about the PU presence. As a result, a SU may transmit during the presence of the PU or may keep starving for the spectrum. In this paper, we propose a reputation-based mechanism for cooperative spectrum sensing which can minimize the effects of MSUs on decision making. Some of the SUs are selected as distributed fusion centers (DFCs), which are responsible for making decisions about the PU presence and inform the reporting SUs. A DFC uses weighted majority voting among the reporting SUs, where weights are determined based on 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 get high reputations. We provide extensive simulations to validate our proposed model.**

Index Terms*—*SSDF, spectrum sensing, cognitive radio networks, spectrum sensing data falsifying attack, security, spectrum security

I. Introduction

In a cognitive radio network (CRN), users 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 SU detects the channel to be free (false-negative), or the PU is not transmitting but SU detects the channel to be occupied (false-positive). A false-negative scenario leads a SU to transmit and cause interference with PU, which is unexpected. A false-positive scenario prevents 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. A SU determines the channel status based on its own sensing information and others' sensing information. There are two types of architecture 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 SUs send sensing information to the fusion center (FC), and a FC makes a decision. SUs ask the FC for the channel status before starting transmissions.

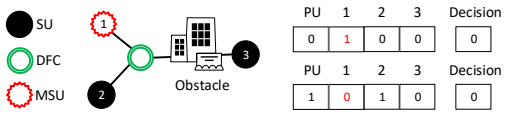


Fig. 1. Example of an SSDF attack.

We consider that there are some malicious SUs (MSUs) in the system, but their population is not more than that of the benign SUs. MSUs send incorrect sensing information to the FC or to other SUs to change the results. This kind of attacks are called spectrum sensing data falsifying (SSDF) attack. If the majority of SUs are benign, then the majority vote gives the correct result in most cases. Sometimes MSUs and the benign SUs with wrong sensing results will win the vote. In Fig. 1, SU 3 remains behind an obstacle. The error rate of SU 3 is greater than other SUs, but less than MSU 1 . We denote the PU presence as 1 and absence as 0. When the PU is absent, the sensing results of 1, 2, and 3 are 1, 0, and 0, respectively. The majority decision is 0, which is correct. When the PU is present, the sensing results of 1, 2, and 3 are 0, 1, and 0, respectively. The majority decision is 0, which is incorrect. Therefore, it is important to detect the MSUs and reduce their weights in election.

In this paper, we propose an online learning-based algorithm to calculate the reputation of SUs at the distributed fusion center (DFC). We use an adaptive learning rate based on the confidence. We consider the confidence of an election as the difference between the population of the majority and minority. When the confidence level is high, reputation increases or decreases at a higher rate. When the confidence level is low, reputation increases or decreases at a lower rate. This adaptive learning rate helps the system identity the MSUs correctly and quickly. We conduct extensive simulations to compare our proposed cooperative spectrum sensing scheme with some existing schemes.

The remainder of the paper is as follows: Section II describes some related works. Section III describes the system and the attacker model. Our proposed SSDF mitigation scheme is presented in Section IV. Some existing and proposed reputation calculation schemes are presented in Section V. Section VI presents rules for combining other DFCs decisions with other DFCs' observations. In Section VII, experimental results are presented.

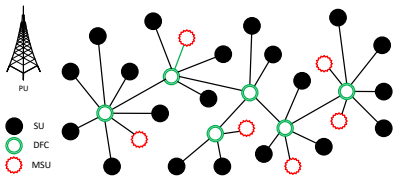


Fig. 2. Cooperative sensing system.

II. Related Work

There are many existing works on cooperative sensing under the SSDF attack. Solutions are based on the authentication of SUs [2], clustering 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 a SU's sensing result is different from (or the same as) the majority, then its reputation is decreased (or increased) by one.

A PU emulation-based testing scheme, FastProbe, creates PU signals to test whether the SUs are reporting honestly or not [11]. They propose a scheduling algorithm to test the SUs periodically. This detection technique is now ineffective, because there are a lot of mechanisms to detect PU emulation signals [12, 13]. In addition, these 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.

III. System and Attacker Model

In this section, we define the attacker and the system model.

A. System Model

We consider a CRN with some SUs and a PU. The PU goes on and off frequently 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 to the DFC directly. 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.

A 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 for broadcasting 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 [14] present a node-degree-based dominating node selection process. A marking process-based CDS construction is proposed in [15]. An energy change-based CDS formation is proposed in [16]. The reputation of a SU can be used as weight in this scheme which will prevent the MSUs from becoming a DFC. 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.

B. Attacker Model

The DFCs only know the SUs who send sensing information to them. A DFC 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. Based on the attacker's behavior, we classify the attacking strategies into four classes:

a) "Random Yes" Attack: The MSU sends "1" to the DFC regardless of the sensing result with α probability. When $\alpha = 1$, the MSU always sends "1" to the DFC; this is called the "Always Yes" attack.

b) "Random No" Attack: This attack is just the opposite of the "Random Yes" attack. The MSU sends "0" to the DFC regardless of the sensing result with α probability. When $\alpha = 1$, the MSU always sends "0" to the DFC; this is called the "Always No" attack.

c) "Random False" Attack: The MSU sends an opposite sensing result to the DFC with a probability of α, meaning that when the MSU's sensing result is "1", it sends "0", and when it is "0", it sends "1" to the DFC. $\alpha = 1$ means that the MSU always sends the opposite sensing result.

d) "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 timeslot with α probability. $\alpha = 1$ means that the MSU sends a random sensing result in each timeslot.

IV. 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 goes 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. 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 of vote, we conclude that the confidence 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 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 a suitable WUF. We discuss some WUFs in the next section.

Algorithm 1 Online Learning-Based Spectrum Sensing

Initialize reputation $r_0[i]$ for SU i for $i \in N(DFC)$ and $t \leftarrow 0$.
while Receive D_t from neighbors where $D_t[i]$ denotes sensing result of SU i at time t **do**
  $C_0 \leftarrow$ number of 0s in D_t and $C_1 \leftarrow$ number of 1s in D_t.
  if $\sum_{i=0}^{I} w_i D_t[i] > Th$ **then**
    $PU \leftarrow 1$.
  else
    $PU \leftarrow 0$.
  confidence $\rho_t \leftarrow \left| \frac{C_0 - C_1}{C_0 + C_1} \right|$
  $x_t[i] = \begin{cases} 1, & \text{if } D_t[i] = PU \\ -1, & \text{if } D_t[i] \neq PU \end{cases}$
  $r_{t+1}[i] = f(*)$
  $w_{t+1}[i] = \frac{r_{t+1}[i]}{\sum_{i=0}^{I} r_{t+1}[i]}$
  $t \leftarrow t + 1$.

V. 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.

A. Linear WUF

Some articles like [10, 17] 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 means an SU's own observed information. Second-hand information means 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 reduces.

B. 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:

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

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.

C. 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 a 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 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:

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

VI. 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 I summarizes this concept.

TABLE I
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

Observing Table I, 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:

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

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.

A. Combining Results from SUs and other DFCs

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.

TABLE II
FC AND DFC COMPARISON

	Fig. 3(f)	Fig. 3(g)
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%]

This adjustment is helpful when a DFC is surrounded by many MSUs. Th'' is another system variable similar to Th and Th'.

B. Performance Analysis

In the worst case, we assume the population of benign the 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 $P(x)$.

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

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.

VII. EXPERIMENTAL RESULTS AND SIMULATIONS

A. Comparison among different WUFs

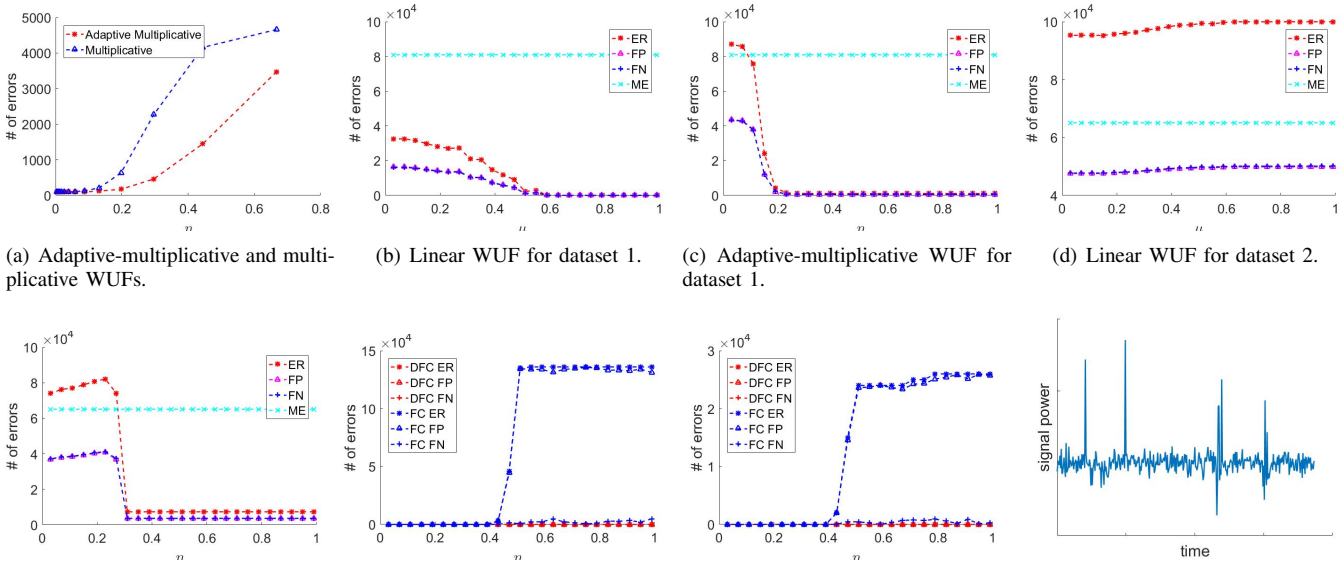
We compare the performances of linear, multiplicative, and adaptive multiplicative WUFs for two datasets. We consider 10 benign SUs with sensing error rates within $[0, 0.3]$ and 10 MSUs with error rates within $[0.8, 1]$. Both datasets have sensing results over $200,000$ timeslots. In dataset 1, the 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.

Fig. 3(a) shows comparison between adaptive multiplicative and multiplicative WUFs. It is observed that the number of errors in adaptive multiplicative WUF is less than that of the multiplicative WUF. Figs. 3(b) and 3(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 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 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. 3(d) and 3(e) show the ER, FP, FN, and ME in the linear and adaptive multiplicative WUFs for different 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.

B. Simulation with real primary user Data

In the experiments above, we consider the PU's presence in each timeslot to be random. To get 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 PU remains OFF for long stretches of the time before suddenly coming ON at some timeslots. Fig. 3(h) 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. 3(f) and 3(g) 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 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. 3(f) and 3(g), 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 the scenario where SUs/MSUs frequently changes behavior. For example, in Fig. 3(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



(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.

(f) FC and DFC based system.

(g) FC and DFC based system.

(h) Received signal power.

Fig. 3. Performance comparison among different WUFs.

the conventional FC-based system. Table II shows detailed parameters of the simulations.

VIII. 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] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.