Low-cost Influence-Limiting Defense against Adversarial Machine Learning Attacks in Cooperative Spectrum Sensing

Zhengping Luo
University of South Florida.
Email: zhengpingluo@usf.edu.

Shangqing Zhao
University of South Florida.
Email: shangqing@usf.edu.

Rui Duan
University of South Florida.
Email: ruiduan@usf.edu.

Zhuo Lu
University of South Florida.
Email: zhuolu@usf.edu.

Yalin E. Sagduyu
Intelligent Automation Inc.
Email: ysgduyu@i-a-i.com.

Jie Xu
University of Miami.
Email: jiexu@miami.edu.

ABSTRACT

Cooperative spectrum sensing aims to improve the reliability of spectrum sensing by individual sensors for better utilization of the scarce spectrum bands, which gives the feasibility for secondary spectrum users to transmit their signals when primary users remain idle. However, there are various vulnerabilities experienced in cooperative spectrum sensing, especially when machine learning techniques are applied. The influence-limiting defense is proposed as a method to defend the data fusion center when a small number of spectrum sensing devices is controlled by an intelligent attacker to send erroneous sensing results. Nonetheless, this defense suffers from a computational complexity problem. In this paper, we propose a low-cost version of the influence-limiting defense and demonstrate that it can decrease the computation cost significantly (the time cost is reduced to less than 20% of the original defense) while still maintaining the same level of defense performance.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**; • **Networks** → Network reliability.

KEYWORDS

Cooperative spectrum sensing, machine learning, adversarial machine learning, attack, defense, data fusion

ACM Reference Format:
Zhengping Luo, Shangqing Zhao, Rui Duan, Zhuo Lu, Yalin E. Sagduyu, and Jie Xu. 2021. Low-cost Influence-Limiting Defense against Adversarial Machine Learning Attacks in Cooperative Spectrum Sensing. In *3rd ACM Workshop on Wireless Security and Machine Learning (WiseML '21), June 28-July 2, 2021, Abu Dhabi, United Arab Emirates.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3468218.3469051

1 INTRODUCTION

Cooperative spectrum sensing is an effective solution to improve the reliability of spectrum sensing and consequently the spectrum utilization through deploying a sensing network that consists of multiple sensing devices and a data fusion center where sensing results are aggregated and a joint decision is made [1]. One real-world example that can benefit from cooperative spectrum sensing is that many of the designated TV spectrum channels are under-utilized leading to the waste of spectrum resources [2].

In the scenario of using cooperative spectrum sensing networks to improve the spectrum utilization. e.g., in TV bands, each of the sensing device in the cooperative spectrum sensing network independently senses the status of a specified (e.g., TV spectrum) channel and sends the sensed signals to the fusion center. The fusion center makes a channel status decision; thus, the secondary users could reference and decide whether to transmit their own signals, or not. The performance of the cooperative spectrum sensing network has a direct impact on how much interference the primary users (e.g., the TV broadcasting stations) will receive.

Cooperative spectrum sensing is susceptible to attacks, where malicious sensors may report erroneous sensing results [3, 4]. Various methods have been proposed to defend the fusion center against such attacks. Statistics-based defense focuses on developing a statistical measure for each sensing node [5–7]. Machine learning-based defense applies machine learning techniques to detect potential malicious nodes [2, 8, 9]. Another type of defense is based on the reputation or trust value of each sensing node. This defense compares the decision of each sensing node with the decision of the fusion center and checks the consistency to defend the fusion center against malicious sensing nodes [10–12].

One typical assumption in existing defenses is that cooperative spectrum sensing network attackers are passive entities, for example, it is assumed that the prior knowledge of the attacks is known to the data fusion center and this knowledge does not change over time [2, 5, 11]. In [13], a machine learning-empowered Learning-Evaluation-Beating (LEB) attack was developed to compromise the sensing network by controlling a small number of sensing nodes. The basic idea of the LEB attack is taking advantage of the black-box nature of the fusion center: the attacker builds its own substitute model of the targeted decision model in the fusion center and (based on this substitute model) crafts adversarial sensing reports sent to the fusion center. During this procedure, the attacker has the capability of "hiding" the behaviors of malicious nodes and achieves its attack utility when vulnerabilities of the decision model are identified.

The LEB attack is built upon adversarial machine learning techniques [14] that have been widely studied to launch attacks on

WiseML '21, June 28-July 2, 2021, Abu Dhabi, United Arab Emirates

ACM ISBN 978-1-4503-8561-9/21/06…$15.00
https://doi.org/10.1145/3468218.3469051

machine learning systems in wireless communications [15–18]. In particular, attacks driven by adversarial machine learning have been considered for spectrum sensing by individual sensors [19–21] and cooperative spectrum sensing [13, 22].

Given the learning-empowered LEB attack and other potential intelligent attacks [4, 22] in cooperative spectrum sensing, [13] provides an influence-limiting defense. By limiting the influence of each sensing node or a subset of nodes toward the final decision, the capabilities of malicious nodes are contained and decreased especially when those nodes are controlled by a "smart" attacker. The main reason that the intelligent attackers could succeed in compromising the fusion center by controlling a small number of malicious nodes is that the attacker could find a more efficient way to hide the malicious behaviors while gradually gaining a high weight in the decision process.

The original influence-limiting defense suffers from a computational complexity problem. In this paper, we offer a solution to decrease the computational complexity of influence-limiting defense. We show that this low-cost version of influence-limiting defense can reduce the computational significantly. On the average, the low-cost version needs less than 20% of the time required by the original influence-limiting defense while still maintaining roughly the same level of defense performance as the original defense.

The rest of the paper is organized as follows. Section 2 defines the system model and describes the LEB attack. Section 3 presents the technical details of influence-limiting defense and its low-cost version. Section 3 describes the experimental configuration and provides the numerical results. Section 5 concludes the paper.

2 PRELIMINARIES

In this section, we define the systems model and describe the LEB attack.

2.1 System Model

Cooperative spectrum sensing provides an effective mechanism to counter the uncertainties experienced by individual spectrum sensing devices [10]. With the availability of low-cost software-defined radio platforms, local and low-cost white space detection mechanisms [23] based on cooperative spectrum sensing open a promising frontier for the easy spectrum usage, especially TV white space, by secondary spectrum users.

The IEEE workgroups have standardized 802.11af (a wireless LAN standard, ranges up to 1 km) [24] and 802.22 (a wireless regional area network standard, ranges up to 100 km) [25] as two white space cognitive radio standards. While dynamic spectrum sensing property is not required in 802.11af, spectrum sensing capability is included as a mandatory feature in the 802.22 standard for cognitive Wireless Regional Area Networks (WRANs) to identify the presence and the type of incumbent users.

We consider a centralized cooperative spectrum sensing model in this paper, which consists of several sensing nodes and a data fusion center. The fusion center makes the channel usability decision based on the inputs from all sensing nodes. Table 1 provides the mathematical notations used in this paper. The sensing result vector $\mathbf{x}$ received by the data fusion center is formed by $x_i, 0 < i \leq n$, as

$$\mathbf{x} = [x_1, x_2, ..., x_n]^\top, \mathbf{x} \in \mathcal{X}, \mathcal{X} \subset \mathbb{R}^{n\times 1}, \quad (1)$$

Table 1: Main notations used in this paper.

O	Targeted decision function in data fusion center.
n	Number of total sensing nodes.
m	Number of manipulated sensing nodes.
x_i	Sensed value of ith node.
$\mathbf{x}$	Sensed signal vector of all nodes.
$\mathbf{x}^*$	Adversarial version of $\mathbf{x}$.
$\mathbf{a}$	Sensed signal vector for manipulated nodes.
$\mathbf{a}^*$	Adversarial version of $\mathbf{a}$.

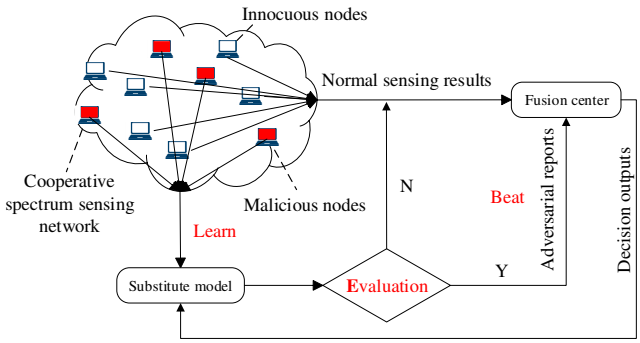

Figure 1: The LEB attack framework [13].

where x_i is the sensing result of the ith sensing node, n is the total number of sensing nodes, and $\mathcal{X}$ is the sensing result space. The data fusion center in cooperative spectrum sensing makes the final channel status decision y based on the sensed result $\mathbf{x}$. We denote the decision mapping function used in the targeted fusion center as

$$O : \mathcal{X} \rightarrow \mathcal{Y}, \mathcal{X} \subset \mathbb{R}^{n\times 1}, \mathcal{Y} = \{-1, 1\}, \quad (2)$$

where $\mathcal{Y}$ is the decision output space with -1 (1) denoting the channel available (unavailable). Among n sensing nodes, we assume that there are m nodes manipulated by an attacker. The data fusion center in the targeted cooperative spectrum sensing framework is treated as a black box to the attacker.

2.2 LEB attack

The LEB attack is empowered by adversarial machine learning. There are two main points inspiring the design of the LEB attack in [13]: (i) the fusion center receives the reports from all sensing nodes to make a final decision and then broadcasts the result to each node; (ii) each sensing node within the network performs the same task independently and the fusion center aims to achieve a better decision performance based on the decision of each node. The basic framework of the LEB attack is shown in Figure 1. In the LEB attack, the fusion center is treated as a black box to the attacker and the attacker only has control of a small number ($m < \frac{n}{2}$) of sensing nodes. The LEB attack is performed in three steps:

- ***Learning*** Inspired by no free lunch theorem [26] and transferability [27] of machine learning, the attacker learns a surrogate machine learning model set, each of the sub-model within the model set is a mirror model of the targeted decision model in the fusion center.

- ***Evaluation*** The attacker evaluates the performance of the sub-models and then chooses the optimal sub-model as the final surrogate model of the targeted decision model. Also, it evaluates the crafted adversarial results in the beating process and decides on whether to launch the attack.
- ***Beating*** Based on the learned optimal surrogate model, the attacker crafts adversarial sensing reports and sends it to the fusion center if it decides to launch the attack; otherwise, it sends the original sensing results to the fusion center.

The main reason that the learning empowered LEB attack can reduce the performance of fusion center by controlling only a small number of sensing nodes is that the LEB attack offers a strategy to identify the potentially vulnerable sensing timeslot and provides an efficient method to generate malicious reports, thus hiding malicious behaviors of the controlled nodes in a "smart" way.

3 INFLUENCE-LIMITING AS A DEFENSE

In this section, we detail the influence-limiting defense [13] and offer our low-cost version of the defense method.

3.1 Influence-limiting defense method

For defense mechanisms against intelligent attacks such as the LEB attacks, there are three main aspects to consider: (i) the wireless nature of the cooperative spectrum sensing could lead to uncertainty of the signals sensed at each node; (ii) the uncertainties of the performance of controlled nodes could be mitigated based on signals sensed by other controlled nodes and therefore a reasonable attack budget could be maintained for each controlled node; and (iii) the adversarial sensing reports are created carefully such that the pattern deviation of each controlled node could be minimized.

To quantify the impact, weight, or influence of each sensing node towards the decision output, a measure named as decision-flipping influence was proposed in [13]. Decision-flipping influence I of a sensing node or $I(\mathcal{X}_{\text{sub}})$ of a subset of sensing nodes is the estimation of the probability that the decision of the fusion center will change given different sensing reports of $\mathcal{X}_{\text{sub}}$. Mathematically, it is estimated through

$$I(\mathcal{X}_{\text{sub}}) \quad \hat{=} \quad \frac{\text{number of } \mathbf{a}^*}{\text{number of } \mathbf{a}}, \text{ subject to } O(\mathbf{x}^*) \neq O(\mathbf{x}), \tag{3}$$

where $\mathbf{a}$ is the sensing result of manipulated sensing devices and $\mathbf{a}^*$ is the manipulated result of $\mathbf{a}$. (3) defines that the decision-flipping influence $I(\mathcal{X}_{\text{sub}})$ is the probability of finding a $\mathbf{a}^*$ based on $\mathbf{a}$ that will change the decision output in the fusion center, which is a direct numerical measure of the influence or impact the subset of nodes $\mathcal{X}_{\text{sub}}$ has towards the decision output in the fusion center.

Given the defined decision-flipping influence $I(\mathcal{X}_{\text{sub}})$, the optimization problem for the influence-limiting defense is formulated in [13] as

$$\begin{aligned} &\text{minimize: } (y - \hat{y})^2, \\ &\text{subject to: } I(\mathcal{X}_{\text{sub}}) \leq \delta(|X_{\text{sub}}|), \forall \mathcal{X}_{\text{sub}} \subset \mathcal{X}, \end{aligned} \tag{4}$$

where y and $\hat{y}$ are the true channel status and the decision output of the fusion center, respectively, $\delta(|X_{\text{sub}}|)$ denotes the cap function, which limits the maximum value of the decision-flipping influence of each subset $\mathcal{X}_{\text{sub}}$, and $|X_{\text{sub}}|$ is the size of $\mathcal{X}_{\text{sub}}$.

The choice of the cap function $\delta(|X_{\text{sub}}|)$ is critical in defending the fusion center. There are two factors to be considered when choosing $\delta(|X_{\text{sub}}|)$: (i) In a general scenario where no malicious node is present, the cap function $\delta(|X_{\text{sub}}|)$ should not have much interference with the decision process, i.e., the performance of the fusion center should not be reduced much. (ii) In scenarios where malicious nodes might exist, the value of $\delta(|X_{\text{sub}}|)$ should be contained under a restricted threshold based on the statistical property of the nodes to limit the attack capability.

Based on these two factors, there is a balance between the performance and the security. If the cap function is defined too restrictive, there will be a higher performance cost. Thus, the cap function given in [13] offers the user with two parameters to balance different security or performance requirements. The cap function is formulated as

$$\begin{aligned} \delta(|\mathcal{X}_{\text{sub}}|) = \frac{1}{1 + e^{-c_1(|\mathcal{X}_{\text{sub}}| - \frac{n}{2})}} - c_2 \sum_{i \in \mathcal{X}_{\text{sub}}} d_{\text{ks}}^i, \\ 0 \leq |\mathcal{X}_{\text{sub}}| \leq \frac{n}{2}, 0 < \delta(|\mathcal{X}_{\text{sub}}|) < 1, \end{aligned} \tag{5}$$

where c_1 and c_2 are the cost control parameter and influence control parameter, respectively, and $\sum_{i \in \mathcal{X}_{\text{sub}}} d_{\text{ks}}^i$ is the sum of Kolmogorov-Smirnov statistics of sensing nodes with the fusion center. In practice, we just need to check the weight or the decision flipping influence of $\mathcal{X}_{\text{sub}}$. If it is below the value defined by the cap function, we can proceed without any modification; otherwise. we need to limit and check the status of sensing nodes within $\mathcal{X}_{\text{sub}}$.

3.2 Computational cost analysis and our optimization

In the influence-limiting defense, the core idea is to contain or limit the decision-flipping influence of $\mathcal{X}_{\text{sub}}$ towards the decision process in the fusion center. However, in practice, to fully limit the attack power of all malicious nodes, it is necessary to enforce the influence-limiting defense at a different level in terms of the size of $\mathcal{X}_{\text{sub}}$. The defender does not usually know how many nodes in its cooperative spectrum sensing network are controlled by the attacker. Note that the total number of subsets in $\mathcal{X}$ with n nodes is $\sum_{k=1}^{n} \binom{n}{k}$, which means the computational complexity in terms of subset will be at the level of $O(2^n)$. This might make it difficult for the fusion center to enforce a full version of influence-limiting defense. Therefore, another parameter η was introduced in [13] to control the complexity.

From a practical point of view, we do not need to evaluate all the combinations of $\mathcal{X}_{\text{sub}}$ from $\mathcal{X}$. For example, if we find that the decision-flipping influence for a specified subset $\mathcal{X}_{\text{sub}}$ is below the value defined by the cap function, then we actually do not need to further examine all the subsets within $\mathcal{X}_{\text{sub}}$, which will reduce the computational cost of enforcing influence-limiting defense significantly. Therefore, the problem turns out to be a challenge to find the most suspicious subset of $\mathcal{X}_{\text{sub}}$ to enforce the full version of influence-limiting policy within that subset. Inspired from the "divide and conquer" scheme in traditional algorithm design, we propose a low-cost version of influence-limiting defense.

We employ the strategy of a top-down style, in which we first divide all the sensing nodes into two subsets randomly and then evaluate the decision-flipping influence $I(\mathcal{X}_{\text{sub}})$ of each subset. In

a normal scenario where no malicious nodes are present and each innocuous node is also well-behaving, the decision-flipping influence for each subset should be very similar with each other and well-balanced. However, if there exist malicious nodes in either subset or in both subsets, the decision-flipping influence for the subset that has the larger number of malicious nodes will have a higher decision-flipping influence if the attacker is successful. If our cost control parameter c_1 and influence control parameter c_2 are chosen appropriately, the proposed influence-limiting defense should be triggered on that subset. Then, in the next round of search, we focus on the subset that triggers the influence-limiting defense. We perform the second round of "divide" operation on the subset, divide it into two smaller subsets and evaluate the decision-flipping influence of each subset again. We will continue the search iteratively by dividing the chosen subset.

At the end of "divide and conquer" search process, we will reach a point where both the two subsets trigger the influence-limiting defense, or at least one subset is non-divisible if there exist malicious nodes like LEB attackers. There is a possibility that both of the subsets include enough malicious nodes to make their decision-flipping influence larger than the value defined by the cap function $\delta(|\mathcal{X}_{\text{sub}}|)$. If there is no malicious node, it will be very difficult to trigger the defense based on the definition of $\delta(|\mathcal{X}_{\text{sub}}|)$.

The key point of this "divide and conquer" strategy is that we do not need to enforce the influence-limiting defense until we reach the point where both the subsets trigger the defense. Before that point, all we need to do is to evaluate the decision-flipping influence $\mathcal{I}(\mathcal{X}_{\text{sub}})$, which will decrease the computational cost of enforcing influence-limiting defense significantly. The detailed process of our low-cost version of influence-limiting defense is shown in Algorithm 1. The advantage of the proposed low-cost version of influence-limiting defense is that it divides the set of the sensing nodes evenly (or roughly evenly if the number of nodes is an odd number) into two subsets. Thus, it will be very efficient to locate the subset that reaches the cap function if there are malicious nodes.

Algorithm 1: Low-cost influence-limiting defense

Input : Historical sensing results and the corresponding decision outputs in $\mathcal{Y}$; parameters c_1, c_2.

Divide $\mathcal{X}$ evenly into two subsets $\mathcal{X}^1_{\text{sub}}, \mathcal{X}^2_{\text{sub}}$;
do :
  If any of the two subsets is an empty set:
    Compute $I(\mathcal{X}_{\text{sub}})$ of the non-empty subet and
    enforce influence if it is larger than $\delta(\mathcal{X}_{\text{sub}})$;
    otherwise terminate the defense process.
  else:
    Compute $I(\mathcal{X}^1_{\text{sub}}), I(\mathcal{X}^2_{\text{sub}})$;
  If $I(\mathcal{X}^1_{\text{sub}}) < \delta(|\mathcal{X}^1_{\text{sub}}|)$ and $I(\mathcal{X}^2_{\text{sub}}) < \delta(|\mathcal{X}^2_{\text{sub}}|)$:
    terminate, no malicious subset is detected;
  If $I(\mathcal{X}^1_{\text{sub}}) \geq \delta(|\mathcal{X}^1_{\text{sub}}|)$ and $I(\mathcal{X}^2_{\text{sub}}) < \delta(|\mathcal{X}^2_{\text{sub}}|)$:
    Update the two subsets $\mathcal{X}^1_{\text{sub}}, \mathcal{X}^2_{\text{sub}}$ with two new
    evenly divided subsets from $I(\mathcal{X}^1_{\text{sub}})$;
  If $I(\mathcal{X}^1_{\text{sub}}) < \delta(|\mathcal{X}^1_{\text{sub}}|)$ and $I(\mathcal{X}^2_{\text{sub}}) \geq \delta(|\mathcal{X}^2_{\text{sub}}|)$:
    Update the two subsets $\mathcal{X}^1_{\text{sub}}, \mathcal{X}^2_{\text{sub}}$ with two new
    evenly divided subsets from $I(\mathcal{X}^2_{\text{sub}})$;
  If $I(\mathcal{X}^1_{\text{sub}}) \geq \delta(|\mathcal{X}^1_{\text{sub}}|)$ and $I(\mathcal{X}^2_{\text{sub}}) \geq \delta(|\mathcal{X}^2_{\text{sub}}|)$:
    Enforce influence through limiting the weight
    $w(\mathcal{X}^1_{\text{sub}}), w(\mathcal{X}^2_{\text{sub}})$;
until $\mathcal{X}^1_{\text{sub}}, \mathcal{X}^2_{\text{sub}}$ are both are empty sets:

4 EXPERIMENTAL PERFORMANCE COMPARISON AND ANALYSIS

In this section, we comprehensively evaluate the performance of the proposed defense and compare it with the original influence-limiting defense from [13].

4.1 Experimental configuration

To fully compare the performance of the original influence-limiting defense and our low-cost version of influence-limiting defense, we employ the same experimental configuration in [13], in which we have 20 sensing nodes in the cooperative spectrum sensing network, and vary the number of malicious nodes to fully demonstrate the defense performance under attacks of different powers. We also focus on defending against the LEB attack using the same dataset as collected in [13]. In the configuration of the fusion center, we consider eight existing representative intrusion detection defenses and using Support Vector Machine (SVM) algorithm as the fusion rule. The configuration for the LEB attacker is also the same as that in [13]. The performance metric that we use in our comparison is the *overall disruption ratio*, which is defined as the ratio of successful attacks over the total elapsed timeslots (in each timeslot, a sensing node will report a sensing result to the fusion center):

$$\text{Overall disruption ratio} = \frac{\text{number of successful attacks}}{\text{number of elapsed timeslots}}. \quad (6)$$

4.2 Results and analysis

In our experiments, we first provide the results of our low-cost influence-limiting defense against the LEB attack with different number of nodes controlled. We adopt the same parameter configuration as that in [13], in which the cost control parameter c_1 is set as 0.05 and the influence control parameter c_2 is set as 0.5. We measure the performance of our low-cost influence-limiting defense and compare it to the original version under different number of malicious nodes. The results are shown in Table 2, from which we observe that when there is no LEB attack, the performance cost of our low-cost version is slightly better than the original version of influence-limiting defense. However, as the number of malicious nodes increases, our proposed defense version performs slightly worse than the original version. It is reasonable since when there are malicious nodes in different subsets, there is a small chance of not limiting the attack power of those malicious nodes that exist in the subset where they do not reach the threshold defined by the cap function δ.

In the second group of experiments, we evaluate the overall disruption ratio of the low-cost influence-limiting defense together with different existing defense mechanisms in the fusion center. The number of malicious nodes is set as $m = 8$ for the LEB attack. The

Table 2: Performance comparison regarding overall disruption ratios of two versions of influence-limiting defense with different number of malicious nodes m.

	m				
	0	**2**	**4**	**6**	**8**
Originial version[13]	0.008	0.013	0.050	0.082	0.095
Proposed low-cost version	0.007	0.014	0.043	0.085	0.105

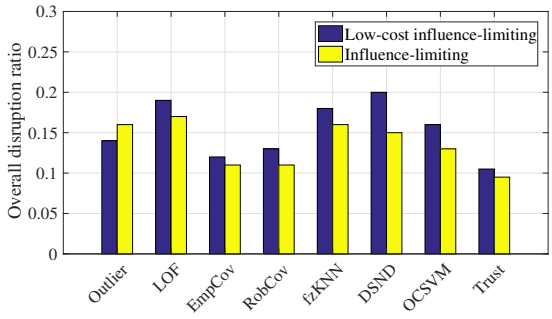

Figure 2: The overall disruption ratio comparison of two versions of influence-limiting policy on different defenses.

Table 3: The normalized average decision time cost comparison of two versions of influence-limiting defense under different number of malicious nodes m.

m	Original version [13]	Proposed low-cost version
0	1.00	0.10
2	1.06	0.11
4	1.08	0.15
6	1.09	0.16
8	1.11	0.18
10	1.12	0.19

results are shown in Figure 2, from which we observe that the low-cost version performs slightly worse than the original version of the influence-limiting defense. The reason is similar to that observed in the first group of experiments. As we set $m = 8$, some malicious nodes do not get their weights limited by the defense.

In the third group of experiments, we compare the complexity cost of the low-cost version of the influence-limiting defense with the original version. In Table 3, we compare the averaged decision time needed when deploying the two versions of influence-limiting defenses. We use a normalized time cost to compare the two methods and set the time cost for the original version of influence-limiting defense as 1 when the number of malicious nodes is 0. The results show that the time cost for our proposed low-cost version of influence-limiting defense is less than 20% of the time cost when running the original version of the defense, which demonstrates the great advantage regarding low complexity of the proposed low-cost version of influence-limiting defense.

In the fourth group of experiments, we examine the normalized time cost regarding different numbers of malicious nodes. We set

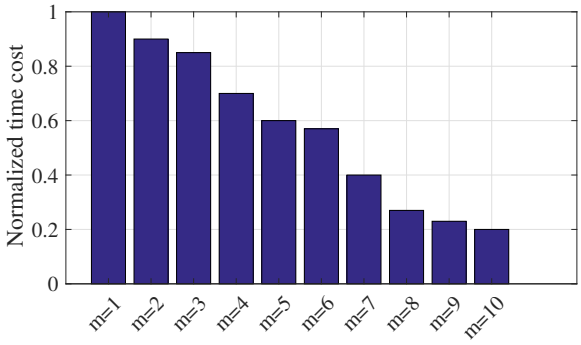

Figure 3: The normalized time cost with regard to different number of malicious nodes when running the low-cost influence-limiting defense.

the time cost when there exists only one single malicious node as 1. The results are shown in Figure 3, from which we can observe that when malicious nodes cover the half of all the sensing nodes, the time cost is only around 20% compared to the scenario when the number of malicious nodes is 1. The reason is that in the low-cost influence-limiting defense, when the number of malicious nodes increases, their attack capabilities accumulate more quickly under the same configuration. Thus, it becomes easier to reach the threshold defined by the cap function $\delta(|X_{\text{sub}}|)$. Further, it becomes easier to trigger the influence-limiting defense, enforcing the weight limitation of $\mathcal{X}_{\text{sub}}$. When the number of malicious nodes is small, it is necessary to examine more rounds of "divide" and evaluate the decision-flipping influence of the corresponding subsets, which requires more time.

The proposed low-cost influence-limiting defense has comparable performance and has much lower complexity. It only incurs a very modest defense performance loss while achieving much better run time efficiency. However, there is still a room to improve. For example, is there any better solution to do the "divide"? Can we achieve even better defense performance while decreasing the run time cost significantly? These aspects are identified as future research direction.

5 CONCLUSION

Cooperative spectrum sensing offers a promising solution to improve the spectrum utilization, especially for TV bands. The security of the cooperative spectrum sensing is quintessential to both primary and secondary users. Influence-limiting defense is a straightforward method to secure the decision process of the fusion center. However, this method suffers a computational cost problem for the real-world applications. In this paper, we presented a low-cost version of the influence-limiting defense by employing the "divide and conquer" strategy as it is much easier to locate the subset that has the largest number of malicious nodes to enforce influence-limiting policy. Our results showed that the low-cost version decreases the run time cost of the defense significantly (to less than 20% of the time required by the original version) with only a slight performance loss.

Acknowledgement: The work at USF is supported in part by NSF CNS-2029875. J. Xu's research is supported in part by NSF CNS-2029858.

REFERENCES

[1] Akyildiz, Ian F., Brandon F. Lo, and Ravikumar Balakrishnan. "Cooperative spectrum sensing in cognitive radio networks: A survey." Physical communication 4, no. 1 (2011): 40–62.

[2] Fatemieh, Omid, Ali Farhadi, Ranveer Chandra, and Carl A. Gunter. "Using Classification to Protect the Integrity of Spectrum Measurements in White Space Networks." In *Network and Distributed System Security Symposium (NDSS)*. 2011.

[3] Zhang, Linyuan, Guoru Ding, Qihui Wu, Yulong Zou, Zhu Han, and Jinlong Wang. "Byzantine attack and defense in cognitive radio networks: A survey." *IEEE Communications Surveys & Tutorials* 17, no. 3 (2015): 1342–1363.

[4] Wang, Wei, Lin Chen, Kang G. Shin, and Lingjie Duan. "Secure cooperative spectrum sensing and access against intelligent malicious behaviors." In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 1267–1275. IEEE, 2014.

[5] Kaligineedi, Praveen, Majid Khabbazian, and Vijay K. Bhargava. "Malicious user detection in a cognitive radio cooperative sensing system." *IEEE Transactions on Wireless Communications* 9, no. 8 (2010): 2488–2497.

[6] Chen, Huifang, Ming Zhou, Lei Xie, and Jie Li. "Cooperative spectrum sensing with M-ary quantized data in cognitive radio networks under SSDF attacks." *IEEE Transactions on Wireless Communications* 16, no. 8 (2017): 5244–5257.

[7] Chen, Changlong, Min Song, Chunsheng Xin, and Mansoor Alam. "A robust malicious user detection scheme in cooperative spectrum sensing." In *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 4856–4861. IEEE, 2012.

[8] Li, Husheng, and Zhu Han. "Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks." *IEEE Transactions on Wireless Communications* 9, no. 11 (2010): 3554–3565.

[9] Thilina, Karaputugala Madushan, Kae Won Choi, Nazmus Saquib, and Ekram Hossain. "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks." *IEEE Journal on Selected Areas in Communications*, 31, no. 11 (2013): 2209–2221.

[10] Chen, Ruiliang, J-M. Park, and Kaigui Bian. "Robust distributed spectrum sensing in cognitive radio networks." In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pp. 1876–1884. IEEE, 2008.

[11] Sagduyu, Yalin E. "Securing cognitive radio networks with dynamic trust against spectrum sensing data falsification." In *2014 IEEE Military Communications Conference*, pp. 235–241. IEEE, 2014.

[12] Rawat, Ankit Singh, Priyank Anand, Hao Chen, and Pramod K. Varshney. "Collaborative spectrum sensing in the presence of Byzantine attacks in cognitive radio networks." *IEEE Transactions on Signal Processing* 59, no. 2 (2010): 774–786.

[13] Luo, Zhengping, Shangqing Zhao, Zhuo Lu, Jie Xu, and Yalin Sagduyu. "When attackers meet AI: Learning-empowered attacks in cooperative spectrum sensing." *IEEE Transactions on Mobile Computing* (2020).

[14] Vorobeychik, Yevgeniy, and Murat Kantarcioglu. "Adversarial machine learning." *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12, no. 3 (2018): 1–169.

[15] Adesina, Damilola, Chung-Chu Hsieh, Yalin E. Sagduyu, and Lijun Qian. "Adversarial machine learning in wireless communications using RF data: a review." *arXiv preprint* arXiv:2012.14392 (2020).

[16] Sagduyu, Yalin E., Yi Shi, Tugba Erpek, William Headley, Bryse Flowers, George Stantchev, and Zhuo Lu. "When wireless security meets machine learning: motivation, challenges, and research Directions." *arXiv preprint* arXiv:2001.08883 (2020).

[17] Erpek, Tugba, Tim O'Shea, Yalin E. Sagduyu, Yi Shi, and T. Charles Clancy. "Deep learning for wireless communications." in Development and Analysis of Deep Learning Architectures, Springer (2020).

[18] Erpek, Tugba, Yalin E. Sagduyu, and Yi Shi. "Deep learning for launching and mitigating wireless jamming attacks." *IEEE Transactions on Cognitive Communications and Networking* 5, no. 1 (2018): 2-14.

[19] Shi, Yi, Tugba Erpek, Yalin E Sagduyu, and Jason Li. "Spectrum data poisoning with adversarial deep learning." In *IEEE Military Communications Conference (MILCOM)*, pp. 407–412. IEEE, 2018.

[20] Sagduyu, Yalin E., Yi Shi, and Tugba Erpek. "IoT network security from the perspective of adversarial deep learning." *IEEE International Conference on Sensing, Communication and Networking (SECON)*, pp. 1–9. IEEE 2019.

[21] Sagduyu, Yalin E., Yi Shi, and Tugba Erpek, "Adversarial deep learning for over-the-air spectrum poisoning attacks," *IEEE Transactions on Mobile Computing*, 20, no. 2 (2021): 306–319.

[22] Luo, Zhengping, Shangqing Zhao, Zhuo Lu, Yalin E. Sagduyu, and Jie Xu. "Adversarial machine learning based partial-model attack in IoT." *ACM Workshop on Wireless Security and Machine Learning (WiseML)*, pp. 13–18. ACM, 2020.

[23] Saeed, Ahmed, Khaled A. Harras, Ellen Zegura, and Mostafa Ammar. "Local and low-cost white space detection." In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 503–516. IEEE, 2017.

[24] Flores, Adriana B., Ryan E. Guerra, Edward W. Knightly, Peter Ecclesine, and Santosh Pandey. "IEEE 802.11 af: A standard for TV white space spectrum sharing." *IEEE Communications Magazine* 51, no. 10 (2013): 92–100.

[25] Stevenson, Carl R., Gerald Chouinard, Zhongding Lei, Wendong Hu, Stephen J. Shellhammer, and Winston Caldwell. "IEEE 802.22: The first cognitive radio wireless regional area network standard." *IEEE Communications Magazine* 47, no. 1 (2009): 130–138.

[26] Wolpert, David H., and William G. Macready. "No free lunch theorems for optimization." *IEEE transactions on evolutionary computation* 1, no. 1 (1997): 67–82.

[27] Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." *arXiv preprint arXiv*:1605.07277 (2016).