When Attackers Meet AI: Learning-empowered Attacks in Cooperative Spectrum Sensing

Zhengping Luo, Shangqing Zhao, Zhuo Lu, Jie Xu, and Yalin E. Sagduyu

Abstract—Defense strategies have been well studied to combat Byzantine attacks that aim to disrupt cooperative spectrum sensing by sending falsified sensing data. However, existing studies usually make network or attack assumptions biased towards the defense (e.g., assuming the prior knowledge of attacks is known). In practice, attackers can adopt any arbitrary behavior and avoid any pre-assumed pattern or assumption used by defense strategies. In this paper, we revisit this traditional security problem and propose a novel learning-empowered framework named Learn-Evaluate-Beat (LEB) to mislead the fusion center. Based on the black-box nature of the fusion center in cooperative spectrum sensing process, our new perspective is to make the adversarial use of machine learning to construct a surrogate model of the fusion center’s decision model. Then, we propose a generic algorithm to create malicious sensing data. Our real-world experiments show that the LEB attack is very effective to beat a wide range of existing defense strategies with an up to 82% of success ratio. Given the gap between the new LEB attack and existing defenses, we introduce a non-invasive and parallel method named as influence-limiting policy sided with existing defenses to defend against the LEB-based or other similar attacks, which demonstrates a strong performance in terms of overall disruption ratio reduction by up to 80% of the LEB attacks.

Index Terms—Cooperative spectrum sensing, system security, defenses and attacks, adversarial machine learning.

1 INTRODUCTION

Cooperative spectrum sensing has been proposed as an effective mechanism to enhance the spectrum sensing performance using cognitive radio devices (e.g., TV-band devices coded in IEEE 802.22). It enables a data fusion-based decision framework, in which multiple nodes report their sensing results to a fusion center that makes a centralized decision to enhance the spectrum sensing accuracy. This, however, opens up opportunities for Byzantine attacks (also widely referred to as spectrum sensing data falsification attacks) [1]–[8], in which attackers aim to send malicious sensing results to mislead the fusion center to make wrong decisions.

Defense mechanisms to combat Byzantine attacks have been widely studied and include: (i) statistics-based mechanisms, i.e., building statistical models to detect or eliminate attacks [4], [9], [11]; (ii) machine learning-based mechanisms, i.e., using machine learning methods as countermeasures [2], [6], [12]–[14]; (iii) trust (or reputation)-based mechanisms, i.e., building trust metrics for nodes to be weighted in the decision process to identify or mitigate attacks [3], [5].

Nevertheless, existing studies mainly make network or attack assumptions that often biased towards defenses. For example, methods used in [4], [9], [11] assume that attacks behave in a particular way or prior information of attack statistics is known such that a statistical model of an attack can be built; and machine learning based methods [2], [6], [13], [14] assume that malicious data pattern deviates from normal data pattern under a given classification rule. These assumptions give an advantage to the defenses over the attacks. However, in practice, attackers can try to actively avoid pre-assumed behaviors or break assumptions used in the defense. Moreover, the time-varying nature of wireless channels and signals can deviate the data properties of the legitimate sensing results from time to time. In this regard, training and prior statistical models used by the defenses face a model mismatch phenomenon over time, which can also be exploited by attackers. All these motivate us to rethink Byzantine attacks from a new perspective: is there a stronger attack model?

Our new observations on cooperative spectrum sensing are twofold: (i) Sensing nodes always report their sensing data as the input to the fusion center; and (ii) regardless of what kinds of defense mechanisms it adopts, the fusion center always announces the final decision, which is the output from the fusion center. If we treat the fusion center as a black box, the defense and decision rules together in the fusion center can be considered as a black-box function with known inputs and outputs, as illustrated in Fig. 1. Inspired by the no free lunch theorem [15] and the transferability [16] property in machine learning, attackers can, in fact, use the inputs and outputs shown in Fig. 1 to build a surrogate model of the targeted fusion center, and then launch effective attacks with minimum data manipulation to mislead data.

Fig. 1: Abstract model of data processing in cooperative spectrum sensing.

Our new observations on cooperative spectrum sensing are twofold: (i) Sensing nodes always report their sensing data as the input to the fusion center; and (ii) regardless of what kinds of defense mechanisms it adopts, the fusion center always announces the final decision, which is the output from the fusion center. If we treat the fusion center as a black box, the defense and decision rules together in the fusion center can be considered as a black-box function with known inputs and outputs, as illustrated in Fig. 1. Inspired by the no free lunch theorem [15] and the transferability [16] property in machine learning, attackers can, in fact, use the inputs and outputs shown in Fig. 1 to build a surrogate model of the targeted fusion center, and then launch effective attacks with minimum data manipulation to mislead data.
the fusion center. However, the model is essentially a partial model of the fusion center.

To make the partial model to attack cooperative spectrum sensing process, we propose a Learn-Evaluate-Beat (LEB) framework in this paper. As its name indicates, LEB consists of three steps: (i) the initial learning step, in which the attacker uses incremental learning models to build its own surrogate model to approximate the fusion center’s decision model; (ii) the evaluation step, in which the attacker evaluates whether its own model is accurate enough or not to launch attacks; and (iii) the final beating step, in which the attacker falsifies the sensing data with the minimum cost to change the fusion center’s decision if they pass the evaluation step. To design LEB attacks to be effective, we also propose a learning algorithm based on a set of sub-models and a generic data generation algorithm to generate adversarial examples (i.e., falsified data with the minimum data manipulation to flip the fusion center’s decision) in sub-models. We conduct comprehensive real-world experiments to measure the performance of spectrum sensing under LEB attacks and their impacts against a wide range of existing defense methods.

Towards the defense perspective, an effective defense mechanism to distinguish LEB-based or similar malicious nodes out from normal nodes is quite challenging. The principal reason that LEB attackers can succeed contributes to that the malicious nodes can build up their influence or impact on the fusion center through taking advantage of the model mismatch phenomenon. We introduce a new metric named as decision flipping influence to measure the influence of a given subset of sensing nodes. Based on the decision flipping influence, we further propose an influence-limiting policy to evaluate and limit the influence any subset of nodes can have on the fusion center, thus decreasing or eliminating the attack capability of the malicious nodes. The influence-limiting policy is a non-invasive, parallel method sided with traditional/existing defenses. Our experimental results demonstrate that the proposed defense can effectively bridge the gap between traditional defenses in cooperative spectrum sensing and the new LEB-based attacks.

Our contributions are listed as follows.

- We rethink the traditional security of cooperative spectrum sensing and present a new perspective to create a stronger attack model named as LEB attack. Our work has shown that the traditional duel of attacks and defenses in cooperative spectrum sensing has to be re-visited in the presence of new learning-based attack models.
- For the LEB framework to be effective and practical, the framework is designed in a flexible way to adopt a wide range of sub-models to build the surrogate model of the fusion center. A generic generation algorithm is proposed to create adversarial examples against cooperative spectrum sensing.
- Our experiments show that the LEB attack can achieve an up to 82% attack success ratio while only manipulating a small number of malicious nodes. The proposed adversarial data generation algorithm achieves similar performances as existing generation methods while reducing by 65% of the computational cost on average.
- We introduce a generic non-invasive method, the influence-limiting policy, sided with traditional/existing defenses to counter LEB attacks or other similar learning-based attacks. Experimental results of influence-limiting policy-based defense demonstrate an average overall disruption ratio reduction by 78% under LEB attacks compared with traditional defenses while without the influence-limiting policy.
- The LEB attack and Influence-limiting policy proposed in this paper can be extended to a more generic partial model problem in machine learning, in which a malicious attacker controls part of the input dimensions to compromise the machine learning model. This paper is a pioneering exploration of the topic in machine learning.

2 MODEL AND PRELIMINARIES

In this section, we introduce the system model, overview existing studies, and identify issues.

2.1 System Model

We consider a cooperative spectrum sensing network scenario with $n$ sensing nodes and one fusion center, as shown in Fig. 2. In each timeslot (i.e., each round of sensing), all nodes perform spectrum sensing on a TV spectrum channel, then report their results to the data fusion center that makes the global channel usability decision based on all inputs. We assume that energy detection [4, 9] is employed at each node, and a sensing report contains the value of the energy level sensed by a node.

The value of the sensed energy level vector at the $i_{th}$ timeslot from $n$ sensing nodes is denoted by $x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]^\top$, $x_i \in \mathcal{X} \subseteq \mathbb{R}^{n \times 1}$, where $\cdot^\top$ is the matrix transpose operator. $\mathcal{X} \subseteq \mathbb{R}^{n \times 1}$ is the value space for the sensed energy level, and $\mathbb{R}^{n \times 1}$ is an $n$-dimensional Euclidean space. The historical sensing results of the sensing nodes are denoted as $\{h_k\}_{k \geq 1, n} = [x_{1,k}, x_{2,k}, \ldots, x_{i,k}]^\top$.

Fusion center makes the final channel status decision $y_i \in \mathcal{Y}$ based on the sensed results $x_i \in \mathcal{X} \subseteq \mathbb{R}^{n \times 1}$. We denote the decision mapping function (implemented by the data fusion rule and the potential defenses) used in fusion center as $O : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} = \{-1, 1\}$ is the decision output space of the fusion center with $-1/1$ denoting the channel available/unavailable. We assume the attacker has no information about the decision mapping function adopted in the fusion center while knows only the final decision $\{y_i\}$ of the fusion center at each timeslot $i$.

Cooperative spectrum sensing enables multiple nodes to report their sensing results to a fusion center to enhance
spectrum sensing accuracy, in which Byzantine attackers can send malicious sensing results to the fusion center, misleading the fusion center to make wrong decisions about the channel status and further cause collisions between primary users of the spectrum with the secondary users.

Without loss of generality, we assume that the first \( m \) nodes in all \( n \) nodes are malicious nodes fully controlled by the attacker. The attacker can make the malicious nodes to report whatever sensing results the attacker wants but has no information about the remaining \( n - m \) innocuous nodes.

We denote the sensing data vector by malicious nodes at timeslot \( i \) as \( \mathbf{a}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,m}]^\top, \mathbf{a}_i \in \mathcal{A} \subset \mathbb{R}^{m \times 1} \), where \( \mathcal{A} \) is the report space of malicious nodes. Obviously, \( \mathbf{a}_i \) is the first part of \( \mathbf{x}_i \), that

\[
\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,m}, x_{i,m+1}, \ldots, x_{i,n}]^\top, \mathbf{x}_i \in \mathcal{X}. \tag{1}
\]

The objective of the attacker is to manipulate sensing data vector \( \mathbf{a}_i \) to mislead the fusion center to yield a wrong sensing decision \( y_i \) for as many timeslots \( i \) as possible.

### 2.2 Defending against Byzantine Attacks

The defenses against Byzantine attacks in cooperative spectrum sensing can be essentially viewed as an anomaly detection problem, in which abnormal nodes need to be distinguished out from innocuous nodes. Most of the defenses proposed in previous works are a combination of two or multiple aspects of node characteristics, such as pattern, statistical consistency or whatever. Here we classify them into three categories by the main node characteristic those defenses used:

- **Statistics-based.** It usually assumes that attackers behave in a particular way or prior information of network or attack statistics are known [4, 9, 11]. As the most widely used approaches to detect malicious nodes, statistics related schemes defend fusion center against attackers based on the statistical measures derived from the sensing report history \( \{\mathbf{h}_k\}_{k \in [1,n]} \), such as covariance, deviation and so on [4, 9, 10].

- **Machine learning-based.** It assumes malicious and innocuous nodes have different underlying data patterns, i.e., \( \{\mathbf{h}_k\}_{k \in [1,n]} \) of malicious nodes and \( \{\mathbf{h}_k\}_{k \in [m+1,n]} \) of innocuous nodes can be classified by machine learning techniques. Both supervised [6] and unsupervised [2, 13, 14] methods have been proposed to identify or eliminate malicious nodes.

- **Trust (or reputation)-based.** The essence of trust based defense is to compute a trust metric based on \( \{\mathbf{h}_k\}_{k \in [1,n]} \). Those nodes with low trust metrics are detected as malicious nodes and get eliminated or less weighted from the decision process. It’s worth noting that trust based methods usually compute trust metrics based on mathematical deviations obtained in statistics based mechanisms [3, 5].

As aforementioned, we find that most of the defenses make various assumptions that are in fact often biased toward the defenses. For example, many of the statistics-based defenses [4, 9] assume that there exist specific statistical models for malicious nodes different from innocuous nodes (e.g., similar distribution among innocuous nodes while different from those malicious nodes); and that different patterns for \( \{\mathbf{h}_k\}_{k \in [1,n]} \) of malicious nodes and \( \{\mathbf{h}_k\}_{k \in [m+1,n]} \) of innocuous nodes can be classified and sorted out.

### 2.3 Model Mismatch in Realistic Network Scenarios

Given the aforementioned assumptions made by the defenses, is there any model mismatch phenomenon between the assumed and realistic scenarios? I.e., is the statistical or other characteristics for each node at different timeslots maintained unchanged? We explore the model mismatch phenomenon both in spatial and temporal dimensions. Two datasets are used in our exploration. One is collected over 13 TV white space channels at 5282 locations in Atlanta metropolitan area [17]. The other one is collected by us using 5 USRP N210s [18] to collect signal strengths at 5 different locations on 22 channels over a \( 20 \times 20 \text{ km}^2 \) urban region for a 100-hour period.

The signal strengths for one TV channel at different locations from the dataset of [17] in Fig. 3(a) illustrates that the sensed signal strengths can be highly related to the location of sensing. Further, we use our dataset to show probability distributions of the TV white space signal strengths at 5 locations in Fig. 3(b). It is easy to observe in Fig. 3(b) that locations C and E have the lowest signal strength while location B has the highest signal strength of one TV channel.

We then measure how the signal strength varies over time. The 100-hour dataset is divided into 5 continuous 20-hour sub-datasets. The probability distributions of the signal strengths for each sub-dataset are plotted in Fig. 3(c), we can see that as time goes by, the probability distribution changes over different sub-datasets. This indicates that a statistical or data model built for one time period will indeed change for another time period.

Our data analysis on existing dataset and collected dataset demonstrate that the statistical/data model mismatch phenomenon over space and time is quite common in practice. In another words, the signal strengths of both malicious and innocuous nodes can hardly follow the exact models/behaviors assumed in a defense strategy. The existence of such phenomenon is also reasonable because of (i) environmental factors, such as weather and buildings, and (ii) network factors, such as co-channel interference and adjacent-channel leakages from other broadcasting activities.

The model mismatch phenomenon will further lead to two consequences: (i) uncertainty of sensing results will occur among multiple nodes (even among innocuous nodes) simultaneously, and (ii) the parameters for the optimal decision to decide the channel status are very hard to find over time. Thus, an intelligent attacker can try to learn what decision models the fusion center uses, then take advantage of the learned model and generate malicious sensing data specifically targeting the decision model.

### 3 LEB Framework: Motivation and Design

In this section, we first present the motivation behind the adversarial learning-based attack design, and then propose the LEB framework against cooperative spectrum sensing.
3.1 Attack Motivation

As we can see, the fusion center can be hard to have perfect models about practical signal data statistics or patterns due to the spatially and temporally varying nature of the wireless environment. Moreover, if we turn the table around and think from an intelligent attacker’s perspective, the attacker should never behave as assumed in any defense.

Our key observations for designing a new attack model are twofold: (i) network nodes always report their sensing data to the fusion center; and (ii) regardless of what kind of defense mechanism it adopts, the center always announces the final decision, which exposes valuable information about the decision model given the input sensing data. If we treat the fusion center as a black box, the decision model (i.e. the defense and the fusion rule) in the center can be considered as a black-box function with known inputs and outputs. As a result, attackers can in fact use the inputs and outputs shown in Fig. 2 to build a surrogate model of the decision model used in the targeted fusion center. After stealing the decision model, the attacker can launch attacks with minimum data manipulation to mislead the fusion center.

Such a learning-based attack model can gradually learn the decision model adopted in the fusion center and is also practically feasible as it does not require the prior knowledge of the decision model or its training data.

3.2 LEB Architecture

Leveraging the new observations, we propose a Learn-Evaluate-Beat (LEB) framework for attacks against cooperative spectrum sensing in this paper. As its name indicates, LEB consists of three steps: (i) the initial learning step, in which the attacker uses incremental learning models to build its own surrogate model to approximate the fusion center’s decision model; (ii) the evaluation setup, in which the attacker evaluates whether its own model is accurate enough to launch attacks, and (iii) the final beating step, in which adversarial sensing results are generated to beat the decision model. The flow chart of the LEB framework is shown in Fig. 4.

3.2.1 Learn

The learning step aims to build the surrogate model $S_i$ at timeslot $i$ to approximate the decision model $O$ in the fusion center given the malicious nodes’ reporting vector $\{a_j\}_{j \in [0, i]}$ and the fusion center’s decision $\{y_j\}_{j \in [0, i]}$. As the LEB attacker has no initial knowledge of the decision model $O$, the learning step in LEB should be generic and flexible.

Our learning idea is inspired by (i) the no free lunch theorem [15], which states that any two machine learning algorithms are equivalent when their performances are averaged across all possible problems, and (ii) transferability [16] in machine learning, which means that as long as models are trained to perform the same task, the influence of adversarial inputs for one model can often be transferred to other models, even if they have different architectures or are trained on different training parameters or datasets.

As a result, we are motivated to use a set of different machine learning models together to approximate the decision model $O$ as the decision model is unknown. Specifically, the surrogate model $S_i$ consists of $L$ machine learning models, called sub-models. Each sub-model is denoted as

\[ S_{i_l} : \mathcal{A} \to \mathcal{Y}, \mathcal{A} \subset \mathbb{R}^{m \times 1}, \mathcal{Y} = \{-1, 1\}, l = 1, 2, \ldots, L, \]

where $m$ is the number of malicious nodes controlled by the attacker. Each sub-model is a particular representative machine learning model and is trained on the same sets $\{a_j\}_{j \in [0, i]}$ and $\{y_j\}_{j \in [0, i]}$.

3.2.2 Evaluate

As Fig. 4 shows, even we use all of the malicious nodes together to build the surrogate model, the model is only a partial one because the fusion center will also use the sensing data from other innocuous nodes. Thus, the goal of the evaluation procedure is to (i) evaluate whether the
partial model is accurate enough and (ii) use model selection to select the best sub-model to launch the attack.

At timeslot \( i \), the attacker uses each of its sub-models \( S_{l,i} \) to classify the current data vector \( a_i \) and obtains its local decision \( z_i = S_{l,i}(a_i) \). Then, the attacker compares \( z_i \) with the fusion center’s decision \( y_i \), and maintains a metric called internal accuracy for each sub-model \( S_{l,i} \), which is defined as \( A_{l,i} = \frac{1}{L} \sum_{j=1}^{L} \mathbf{1}_{\{z_i = y_i\}} \in [0, 1] \), where \( \mathbf{1}_{\{z_i = y_i\}} \) denotes the indicator function that has value of 1 if \( z_i = y_i \) and value of 0 otherwise. The internal accuracy means how accurate a sub-model can track the fusion center’s decision model.

In the evaluation step, if the highest internal accuracy in all sub-models is greater than a given threshold \( \alpha \), the attacker will select the sub-model as its final surrogate model for timeslot \( i \) and enters the beating procedure for the \( i \)-th timeslot. Otherwise, the attacker will not attack, but simply send the real sensing data to the fusion center. This evaluation procedure ensures that the attacker will not attack with low confidence, which is also designed to improve the attack success probability under trust-based defenses.

We denote by \( S_{l^*,i} \) the best sub-model with the highest internal accuracy selected at timeslot \( i \), where \( l^* = \arg\max_{l \in [1, L]} A_{l,i} \) if \( S_{l^*,i} > \alpha \). Then, we can write the surrogate model as

\[
S_{l} = S_{l^*,i} = S_{\arg\max_{l \in [1, L]} A_{l,i}}.
\]

### 3.2.3 Beat

The goal of the beating step is to craft adversarial sensing results based on the selected sub-model to beat the decision model in the fusion center and get a desired output. Denoting by \( a_i^* \) the attacker’s adversarial sensing results, we write \( a_i^* = a_i + \delta_i \), where \( a_i \) is the real sensing data and \( \delta_i \) is called the adversarial perturbation. The beating step finds the best \( \delta_i \) with the minimum data manipulation satisfying

\[
\text{Objective: } \min_{\delta_i \in D} \| \delta_i \|_2,
\]

\[
\text{Subject to: } S_{l^*,i}(a_i^*) = S_{l^*,i}(a_i + \delta_i) \neq S_{l^*,i}(a_i),
\]

where \( \| \cdot \|_2 \) is the L2-norm and \( D \) is the feasible solution space of \( \delta_i \), which is a constraint put onto \( a^* \) to limit the maximum report result \( a_{\text{max}} \) and minimum report result \( a_{\text{min}} \), which ensures that it can avoid being detected under defenses like outlier-based malicious nodes detection \([1]\).

### 3.2.4 Update

All sub-models in the LEB framework need to be updated continuously over time and will be re-trained through adding new input and output data in each round of spectrum sensing. The nature of such re-training is to incrementally add training data to its already trained model, such that a more accurate model can be found as timeslot goes on. Therefore, LEB can adopt online/incremental sub-models to efficiently update its training. In this way, LEB can not only be updated with new sensing data in a dynamic network environment, but also be trained to adopt more information about the fusion center’s decision model.

---

**3.3 Generic Adversarial Sensing Data Generation**

As articulated in the beating step, we must find the best \( \delta_i \) with the minimum data manipulation to satisfy Eq. (4). When the best sub-model is chosen as the surrogate model \( S_i = S_{l^*,i} \), \( \delta_i \) can be found via solving a method-specific optimization problem. For example, we can use the fast gradient sign methods (FGSM) \([19]\) or the Jacobian-based saliency map approach (JBSM) \([20]\) to find the best \( \delta_i \) if the selected sub-model is DNN. However, this complicates the solution to Eq. (4) in the beating step, making it depend on a specific type of sub-models and also makes it less flexible and less generic as the surrogate model contains \( L \) sub-models, which is extendable by design in the LEB framework.

We aim to provide a generic adversarial sensing data generation algorithm to solve Eq. (4). Our design intuition is as follows: (i) Unlike complicated data representation (e.g., image and voice data), signal strength data provides straightforward information: a larger value more likely indicates the fact that a channel is occupied and a less value indicates otherwise. Therefore, search is more likely toward one direction in Eq. (4). (ii) Transferability in machine learning indicates that if we can easily find the adversarial example (i.e., the manipulated input data with the minimum change to make a classifier change its classification output) of one sub-model, we can transfer it to other sub-models. However, in LEB framework, different final surrogate models will be selected as timeslot goes on. Inspired from that, we propose a pilot model-based adversarial sensing result generation method.

The proposed pilot model-based method is specifically designed for the cooperative spectrum sensing scenario, which consists of two steps: Step 1, estimating the decision hyperplane of the training sensing results \( \{a_i\}_{i \in [0, q]} \) from malicious nodes, which is determined by \( \mathbf{w} \cdot a + b = 0 \) for a small set of sensing results \( \{a_i\}_{i \in [0, q]} \) a.k.a. support vectors in SVM \([21]\) from the training data of the surrogate model. \( \mathbf{w} \) is the trained weight vector and \( b \) denotes the bias (or intercept) (as shown in Fig. 5) in which we consider a two dimension situation, denoted as \( x_1 \) and \( x_2 \). Step 2, based on \( \mathbf{w} \) and uses binary search structure along the direction defined by \( \mathbf{w} \) to find the final \( \delta_i \) to form the adversarial report \( a^* \) of the selected sub-model.

The procedure of the proposed method is shown in Algorithm 1. We can use SVM as the pilot model due to its outstanding transferability performances \([16]\) in practise. However, it is not necessary to train an extra pilot model, instead, we can choose the sub-model that has the best internal accuracy with the fusion center as the pilot model.
4 EXPERIMENT EVALUATION

In this section, we collect real-world TV white space signal strengths, based on which we conduct the experiments to measure the impacts of LEB attacks under various conditions.

4.1 Experimental Setup

4.1.1 Measurement Configurations

We collect the realistic TV white space signal strengths based on the RTL-SDR TV dongles, which have been validated to have adequate signal detection capabilities [17]. We deployed 20 RTL-SDR TV dongles as sensing nodes on campus to collect signal strengths on TV channels based on the configurations used in [17] simultaneously. We use GNURadio v3.7.9 to implement the sensing process for each dongle that uses the averaged signal power over a time period of 30 seconds (required by Federal Communications Commission [23]) as the sensing result for one timeslot.

To make the collected signal data comprehend the realistic spectrum sensing scenarios as much as possible, we deployed the sensing devices in various surrounding environments. Fig. 6 shows how 20 RTL-SDR TV dongles are deployed throughout a 379 × 232 ft² building: 8 TV dongles are placed outside of the building and 12 are distributed within the building on different floors and indoor environments. We distribute TV dongles at these various places and environments to denote different spectrum sensing scenarios in practice, in which the deployment way is similar with the VT cognitive radio network testbed [24]. We collect the TV white space channel signal strengths on 22 different channels for 100 continuous hours on each sensing node, such that the dynamic model mismatch phenomenon can be sensed and recorded in the dataset.

4.1.2 Fusion Center Configurations

We implement 8 representative defenses in the fusion center. Specifically, four statistics based defenses: Outlier factors based defense (Outlier) [4], Local Outlier Factor based defense (LOF) [25], Empirical Covariance based (EmpCov) and Robust Covariance based (RobCov) detections [26] and three machine learning based defenses: fuzzy kNN based defense (fkKNN) [13], Double-Sided Neighbor Distance based defense (DSND) [2] and One-class SVM based detection (OCSVM) [14]. We also implement one trust-based detection method (Trust) [5].

There are two steps for the fusion center to make a decision. At step 1, the fusion center accepts sensing results from all nodes, based on which it runs the deployed defense strategy to assess the suspicious level of each node. Given the suspicious level, data from each node is re-calibrated such that potential malicious nodes have less impacts on the fusion process. Step 2 is the data fusion process of the re-calibrated nodes. There are various data fusion rules such as SVM, Logistic Regression (LR), AND, OR and Majority rule. Our experimental results show the LEB attack is effective against different types of data fusion rules. Due to the page limit, we use the SVM-based fusion rule [6, 27] to show the experimental results of the LEB attacker in this section of the experiments.
There is no attack in the training process in our experiments. Among all 22 channels collected, we select the channel that has the largest deviations of the signal strengths to evaluate the LEB framework. We use sensed data of the first 5 hours collected from 20 nodes and the corresponding ground truths of the TV channel status in our local area to build statistical models and serve as training data for the implemented defense methods in the fusion center, the remaining 80 hours of the data is used as validation and test data.

### 4.1.3 LEB Framework Configurations

We implement the LEB framework based on malicious nodes. The surrogate model consists of 6 incremental learning based sub-models: Naive Bayes classifier for multinomial models (MulNB), Perceptron classifier (Per), Linear SVM classifier with stochastic gradient descent training (SGD), Passive Aggressive-I classifier (PA-I), Passive Aggressive-II classifier (AP-II) and Multi-layer Perceptron classifier (MLP).

Without loss of generality, we let the duel of cooperative spectrum sensing and the LEB attacker that controls the malicious nodes starts at timeslot 0 (started from the 6th hour) and the defense has already been trained using the first 5 hours of the collected data. At each timeslot (30 seconds for one timeslot), the LEB attacker will learn and evaluate the surrogate model and possible channel status to launch potential attacks. The fusion center will make a binary decision based on the defense and data fusion method.

In the evaluation step, the internal accuracy threshold $\alpha$ is chosen to be 0.85 unless otherwise specified. In the beating step, the LEB attacker generates adversarial sensing data based on the generic generation algorithm proposed in Section 3.3.

### 4.1.4 Performance Metrics

We evaluate the LEB attacker through two metrics: attack success ratio and overall disruption ratio. The attack success ratio is defined as the ratio of the number of attacks that successfully mislead the fusion center to make the wrong decision to the number of attack attempts. We define the overall disruption ratio as the ratio of the number of successful attacks to the number of elapsed timeslots. A higher attack success ratio does not necessarily mean a higher overall disruption ratio. This is because when the LEB attacker does not pass the evaluation step (e.g., it can have a large threshold $\alpha$ to make an attack attempt succeed with high probability), it will not launch the attack and thus will cause no disruption to the network.

In comparison, the attack success ratio measures the learning and evaluation quality of the LEB framework, and the overall disruption ratio quantifies the performance impact the LEB attacker brings to the entire cooperative sensing network.

### 4.2 Results and Analysis

#### 4.2.1 Attack Impacts on Defense Strategies

We first measure impacts of the LEB attacker on each of the 8 defense strategies used by the fusion center. We randomly select 8 nodes as malicious nodes to be controlled by the LEB attacker from the collected dataset. The LEB attacker updates its internal accuracy for each sub-model to evaluate and beat the fusion center. Fig. 7 shows the attacker’s internal accuracy of sub-models over timeslots against defense strategy (a) Outlier, (b) LOF, (c) EmpCov, (d) RobCov, (e) fzeKNN, (f) DSDN, (g) OC-SVM and (h) Trust. We can observe from Fig. 7 that when the attacker initially starts to learn, the internal accuracy of each sub-model changes drastically; but as the timeslot goes on, the internal accuracy becomes more and more stable. For example, in Fig. 7(c), the accuracy start to remain stable at around 0.85 after 500 timeslots.

Fig. 8 illustrates the attack success ratio and overall disruption ratio of the LEB attacker against each defense method. It is noted from Fig. 8 that the LEB attacker achieves 71%–90% attack success ratios against different methods, which means that the learning and evaluation design in the LEB framework is effective for the attacker to assess its potential capability to launch successful attacks. We can also observe from Fig. 8 that the attacker causes overall disruption ratios by 45%–80%, which indicates that the LEB attacker is able to successfully mislead the fusion center to make wrong decisions for 45%–80% of the time, thereby resulting in severe disruptions.

#### 4.2.2 Impacts of Threshold $\alpha$

A key factor in LEB framework is the internal accuracy threshold $\alpha$. The attacker can only launch attacks if the internal accuracy of a sub-model exceeds $\alpha$. A larger $\alpha$ should lead to a higher attack success ratio but not necessarily a higher overall disruption ratio.

Fig. 9 depicts (a) the attack success ratio and (b) the overall disruption ratio under different values of $\alpha$ (other setups are the same as those in Fig. 8 and results are averaged over 8 defense strategies). We can see from Fig. 9(a) that when $\alpha$ is 0.7 or less, the attack success ratio is around 0.5. As $\alpha$ increases, the attack success ratio also increases, which is because an increased $\alpha$ requirement indicates a higher confidence for the selected sub-model.

However, in terms of the overall disruption ratio, although a smaller $\alpha$ will lead to more attack attempts, it will be easier to activate the defense mechanisms in the fusion center to compromise the malicious nodes, further decrease the overall disruption ratio. The overall disruption ratio increases first and then decreases as the threshold $\alpha$ approaches to 1, as shown in Fig. 9(b). This is because as the threshold $\alpha$ approaches to 1, the number of attack attempts decreases dramatically. Thus although the attack success ratio increases, the overall number of attacks plunged. We observe from our experiments that the optimal threshold can be achieved at around $\alpha = 0.85$ to maximize the overall disruption ratio.

#### 4.2.3 Impacts of the Number of Malicious Nodes

The number of malicious nodes controlled by the LEB attacker is a fundamental factor that affects the attack impact. Table 1 shows the overall disruption ratios against different defenses when the number of randomly selected malicious nodes increases from 2 to 10 (i.e., 10% to 50% of all nodes). We can see from Table 1 that as the malicious nodes approaches 10, which is half of the total sensing nodes, the
Fig. 7: Internal accuracy transitions of the LEB attacker under different defenses.

Fig. 8: The Attack success ratio of LEB attacker with different defense strategies.

Fig. 9: Relationship between $\alpha$ and the attack success ratio of LEB attacker.

overall disruption ratio (averaged over all defenses) reaches 72%, which means that the spectrum sensing is disrupted by the attacker 72% of the elapsed timeslots. The attack impact can still be observed even when the number of malicious nodes is small. For example, 3 malicious nodes (15% of all nodes) can lead to a nearly 20% overall disruption ratio.

It can be concluded from Table 1 that the LEB framework provides an effective attack strategy even when the number of malicious nodes is small.

### 4.2.4 Impacts of Locations of Malicious Nodes

In previous experiments, we always randomly select nodes as malicious ones. We are also interested in whether malicious nodes can bring more impacts to the network if they choose to be at “better” locations. We divide the dongles into 4 groups with 5 in each group. The experiments are conducted by using one group as malicious ones while the others being innocuous.

The results are depicted in Fig. 10, from which we know that when controlling nodes 16–20 (distributed in outside environment) as malicious ones, the overall disruption ratio can achieve 43% averaged over all 8 defenses. However, if nodes 6–10 (distributed in indoor environment) are controlled, the averaged overall disruption ratio is only 17%. Hence, we can conclude that it is critical for the LEB attacker to be at the “right” locations in order to launch more effective attacks.

### 4.2.5 Efficiency of Adversarial Sensing Data Generation

We also compare the performance of the proposed adversarial sensing result generation method with other adversarial sample generation methods: Fast gradient sign method

| # Mal. Nodes: | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---------------|---|---|---|---|---|---|---|---|----|
| Outlier       | 8 | 20| 23| 24| 34| 38| 65| 68| 72 |
| LOF           | 6 | 15| 23| 25| 35| 36| 58| 60| 65 |
| EmpCov        | 9 | 22| 27| 35| 37| 39| 52| 59| 66 |
| RobCov        | 7 | 15| 22| 30| 39| 41| 53| 56| 64 |
| fzKNN         | 10| 21| 27| 45| 45| 49| 78| 80| 82 |
| DSDN          | 11| 23| 31| 44| 46| 47| 80| 85| 86 |
| OCSVM         | 9 | 21| 31| 39| 48| 51| 76| 79| 80 |
| Trust         | 6 | 16| 21| 29| 34| 38| 43| 56| 60 |
| **Average**   | 8 | 19| 28| 34| 40| 42| 63| 69| 72 |
**5 Defense Against LEB Attack**

The LEB attack poses a new security threat to security spectrum sensing. In this section, we propose a mechanism named as influence-limiting policy to combat the LEB or machine learning based attacks. We design influence-limiting as a non-invasive method to the existing defenses. Thus, it can bridge the gap between traditional defenses and the new LEB attacks.

### 5.1 Motivation

In a scenario where the LEB attacker is present, most of the traditional defenses, such as statistics-based defenses [4], [9], [11], machine learning-based defenses [2], [6], [13], [14] and trust value-based methods [3], [5] fall short of the capability due to three main challenges in practice:

- The identified model mismatch phenomenon resulted from spatial and temporal unevenness will lead to dynamic changes of the statistical property at each sensing node.
- When the attacker decides not to launch the attack, manipulated nodes can intentionally behave “normal” through correcting random mistakes by combining all the sensing results of manipulated nodes to make the channel status decision. It improves the statistical consistency of those controlled nodes in no-attack timeslots with the fusion center, such that they can maintain competitive attack budget in attack timeslots to avoid trust value-based defenses.
- When the manipulated nodes decide to launch the attack, they can learn an efficient way through the LEB framework. It minimizes the pattern deviations, which makes the pattern-learning based defenses more difficult to distinguish the pattern differences.

To our best knowledge, there are still lack of robust and generic strategies to combat these challenges in cooperative spectrum sensing. On the other hand, from the machine learning perspective, although there are many strategies to combat adversarial samples in machine learning [33]–[35], most of the defense mechanisms for machine learning systems fall into two aspects: gradient masking [35]–[37] and adversarial sample detection [38].

Gradient masking focuses on hiding the gradient information while adversarial sample detection methods can deny access for those detected inputs. Nevertheless, as the decision boundaries are more or less the same in gradient masking methods, they are still highly vulnerable to black-box transfer attacks [39]. In the dynamic spectrum sensing scenario, the LEB attack essentially is a kind of black-box attacks. Existing adversarial sample detection methods are rather heuristic and usually involve training an attack detector [40] based on the ground truth, which falls within...

---

Fig. 10: Overall disruption ratios when manipulating different nodes.

(FGSM) [19], Jacobian-based saliency map approach (JBSM) [20], DeepFool method (DF) [28], Basic iterative method (Balter) [29], SPSA attack method (SPSA) [30] and Elastic-Net method (EN) [31] in Deep Neural Network, implemented based on CleverHans V2.1.0 [32].

In the experiments, 8 nodes are randomly selected as malicious nodes and use different adversarial data generation methods in the LEB framework to generate malicious data inputs to the fusion center. Fig. 11 shows (a) the attack success ratios and overall disruption ratios under different generation methods and (b) the normalized costs of the generation methods in which we define the normalized cost of a generation method as its computational time to generate the adversarial data vector divided by the computational time of our proposed method to generate the adversarial data (thus our proposed method has a normalized cost of 1).

Fig. 11(a) shows that our proposed method has the similar attack success ratios and overall disruption ratios to other methods. Fig. 11(b) further shows that our method is much more computationally efficient than the other methods. For example, SPSA has a normalized cost of around 4.5, and the overall average computation cost reduction compared to other methods is around 65%.

Fig. 11: Comparison of the adversarial sensing result generation methods.
the traditional defenses in spectrum sensing.

Instead of finding a solution in the machine learning domain that unfortunately has not yet well solved the adversarial example issue in cooperative spectrum sensing. Our objective is to design a robust and generic countermeasure named as influence-limiting policy specifically for the spectrum sensing domain. The method aims to bridge the gap between traditional defense methods and new challenges posed by LEB attackers through introducing the decision flipping influence.

5.2 Influence-Limiting Policy

At the first glance, the defense task in our scenario seems quite challenging in that we must have a rule to differentiate the LEB-based malicious nodes from innocuous nodes. Many traditional methods have been proposed to counter the uncertainties of the sensing results from each node. The metrics of “weight”, “trust value”, or “reputation” are widely used to balance the individual decision of each node and the fusion center decision [1]–[8], [41]. All of these metrics are directly or indirectly based on the statistical consistencies of individual decisions with global decisions. Nevertheless, high consistency does not guarantee high worthiness of trust. As we have detailed in the LEB attack strategy, malicious nodes can purposely maintain a high consistency with the fusion center to avoid being detected.

As a result, we focus on finding the reason why a limited number of malicious nodes can succeed the attack after learning the fusion center’s model. Our key observation is that the malicious nodes can take advantage of the model mismatch phenomenon to build up their dominant role in the decision inference process, i.e., the malicious nodes can accumulate their influence on the fusion center, thus the influence of normal nodes are comparatively decreased. From the defense perspective, although we may not know who are attackers in the network exactly, we can practically enforce to limit the influence of any subset of nodes on the fusion center’s global decision.

5.2.1 Design Framework

To measure such an influence of any given subset of nodes $X_{sub}$, we propose a new metric called decision flipping influence, denoted by $I(X_{sub})$ as the probability that given an actual sensing result, the subset $X_{sub}$ is able to find another feasible sensing result (we say a sensing result is feasible if it is within the allowed signal strength range) leading to the opposite decision from the original one by the fusion center. Apparently, the influence of all nodes should always be 1, i.e., $I(X) = 1$. Unlike “weight”, “trust value” or “reputation”, the decision flipping influence $I(X_{sub})$ is a direct measure of the role $X_{sub}$ played in the final decision process, which can be denoted by the possibility for $X_{sub}$ to change the decision output in the fusion center.

One of the advantages of this definition for $I(X_{sub})$ in spectrum sensing domain is that it can be estimated from the difference of the decision outputs of two extreme cases: In the first situation, original sensing results $x_{original}$ are sent as the input, while in the second situation, both the maximum and minimum sensing results, $x_{max}$ and $x_{min}$, of $X_{sub}$ are sent as the inputs. If any maximum or minimum sensing results lead to different decision outputs than original sensing results, we take it into consideration when estimating the influence $I(X_{sub})$. The flipping decision influence $I(X_{sub})$ is estimated by the percentage of the number of timeslots, in which the decision outputs of the maximum or minimum sensing results are different from the original outputs, among all timeslots of interest.

Starting from the decision flipping influence $I(X_{sub})$, we can mitigate the severe impact of potential LEB attack by enforcing an influence-limiting policy, in which the decision flipping influence of any subset $X_{sub} \subset X$ should satisfy $I(X_{sub}) \leq \delta(|X_{sub}|)$, where $|X_{sub}|$ denotes the number of elements in $X_{sub}$, and $\delta(|X_{sub}|)$ is the threshold function of $|X_{sub}|$, i.e., the influence-limiting policy is triggered only when $I(X_{sub})$ goes beyond $\delta(|X_{sub}|)$. For example, $\delta(1)$ denotes the threshold to limit the influence of each individual node. From the LEB framework we know that as the malicious nodes keep succeeding to flip the decision in the fusion center, their corresponding decision flipping influence $I(X_{sub})$ will also increase. Thus any subset including those nodes will eventually go beyond the preassigned threshold $\delta(|X_{sub}|)$. We can write the influence-limiting policy as an non-invasive, parallel method sided with traditional/existing defenses as follows:

$$
\text{Objective: Minimize the prediction error based on traditional/existing defenses.} \quad (5)
$$

$$
\text{Subject to: } I(X_{sub}) \leq \delta(|X_{sub}|), \forall X_{sub} \subset X.
$$

5.2.2 Choosing the Threshold Function

Naturally, the threshold function of $\delta(|X_{sub}|)$ with regard to $|X_{sub}|$ lies at the heart of the influence-limiting policy. To choose the threshold function, we first discuss simple cases, then present the generic case.

(i) In a well-balanced cooperative spectrum sensing scenario without malicious nodes, the value of $\delta(|X_{sub}|)$ in terms of $|X_{sub}|$ should satisfy the following three basic requirements:

(a) $\delta(|X_{sub}|) \rightarrow 0$ when $|X_{sub}| \rightarrow 0$;
(b) $\delta(|X_{sub}|)$ is monotonically increasing with regard to $|X_{sub}|$;
(c) $I(X_{sub}) \rightarrow \frac{1}{2}$ when $|X_{sub}| \rightarrow \frac{n}{2}$, which is to ensure that the influence over the fusion center is dominated by the majority rather than a small group of sensing nodes. Based on the requirements, a Sigmoid style function comes out to be a good choice to interpolate the function of $\delta(|X_{sub}|)$ with regard to $|X_{sub}|$, i.e.,

$$
\delta(|X_{sub}|) = \frac{1}{1 + e^{-c_{1}(|X_{sub}| - \frac{n}{2})}}, 0 \leq |X_{sub}| \leq \frac{n}{2}, \quad (6)
$$

where $c_{1}$ is the control parameter used to adjust the function to better interpolate various practical scenarios. As we assumed that the number of malicious nodes is less than the number of normal nodes, which is the most commonly discussed attack model [1], [2], [4], [7], [8].

(ii) In a generic scenario where malicious nodes may present, the function $\delta(|X_{sub}|)$ defined in (i) has no mechanism to counter the malicious nodes. It’s intuitive that when malicious nodes are present in $X_{sub}$, $\delta(|X_{sub}|)$ should be limited to a smaller threshold value such that the decision flipping influence can be restrained. How can we know which node is malicious? It is noted that due to temporal or spatial
unevenness, it is difficult to accurately identify which node is indeed malicious by looking at the statistical property of the signal strengths. Therefore, instead of offering a hard decision rule to clearly classify a node into either innocuous or malicious, we design a soft rule to discriminate certain nodes in the final decision by the fusion center.

In particular, we still leverage a node’s signal strength, but only look at the changes of its statistical property. Suppose when a node’s signal strengths exhibit different properties during the training and testing (or decision) phases, there exist the following indications:

- The node may be malicious and manipulate its signal strengths for an effective attack. If this is the case, the node should be at least less weighted (if not eliminated) in the fusion center’s decision.
- The node may be legitimate but its signal property changes due to temporal or spatial unevenness, which further means that the original training data for this node does not reflect its current signal property and thus becomes less useful for the current decision.

In both cases, we should at least less weight the node with changes of its statistical signal property in the final decision. Therefore, we adopt the Kolmogorov-Smirnov (K-S) statistic to quantify such a change for a node. For node $j$, the K-S statistic $d_{ks}^j$ is

\[ d_{ks}^j = \sup_x |F_j^t(x) - F_j^c(x)|, \]

where $F_j^t(x)$ is the training data distribution for node $j$ and $F_j^c(x)$ is the empirical distribution of node $j$’s signal strength input to the fusion center during a given test period, representing its current signal property. $sup(\cdot)$ denotes the maximum value or upper bound.

Based on Eq. (7), we present our generalized influence-limiting policy as

\[ \delta(|X_{\text{sub}}|) = \frac{1}{1 + e^{-c_1(|X_{\text{sub}}| - \frac{\eta}{2})}} - c_2 \sum_{j \in X_{\text{sub}}} d_{ks}^j, \]

where $0 \leq |X_{\text{sub}}| \leq \frac{n}{2}$, $0 < \delta(|X_{\text{sub}}|) < 1$, and $c_1$ is a cost control parameter and $c_2$ is an influence control parameter. The policy ensures that the more abnormal for nodes in $X_{\text{sub}}$, the larger $\sum_{j \in X_{\text{sub}}} d_{ks}^j$ will be and thus smaller the threshold $\delta(|X_{\text{sub}}|)$ will be. When all the nodes in $X_{\text{sub}}$ are normal nodes, $\sum_{j \in X_{\text{sub}}} d_{ks}^j$ is expected to be very small.

5.2.3 Enforcing Influence-Limiting

In the fusion center, the objective of the defense is to mitigate the impact posed by potentially malicious nodes through enforcing the influence-limiting policy, i.e., limit the decision flipping influence of any given $X_{\text{sub}}$ within the range of $[0, \delta(|X_{\text{sub}}|)]$. There are multiple ways to enforce the policy, our solution is to re-weight the sensing results of $X_{\text{sub}}$. Specifically, when the measured decision flipping influence $I(X_{\text{sub}})$ goes beyond $\delta(|X_{\text{sub}}|)$ in a traditional defense method, the influence-limiting policy will limit the weight of $X_{\text{sub}}$ to a threshold. In other words, the weight $w(X_{\text{sub}})$ can be written as

\[ w(X_{\text{sub}}) = \begin{cases} 
\delta(|X_{\text{sub}}|), & I(X_{\text{sub}}) \geq \delta(|X_{\text{sub}}|), \\
\text{original weight}, & \text{Otherwise}.
\end{cases} \]

Algorithm 2: Influence-limiting policy

**Input**: Historical sensing results of $X$ and the corresponding decision outputs in $Y_i$; parameters $c_1, c_2, \eta$;

1. for each $X_{\text{sub}}$ according to the value of $\eta$:
   2. Calculate decision flipping influence $I(X_{\text{sub}})$;
   3. Compute threshold $\delta(|X_{\text{sub}}|)$;
   4. Enforce influence through limiting the weight $w(X_{\text{sub}})$;
5. end for

6. Feedback $w(X_{\text{sub}})$ to traditional/existing defenses.

5.2.4 Balancing Effectiveness and Complexity

It is worth noting that in order to achieve the objective defined in Eq. (5), we have to consider all possible subsets in $X$. It becomes computationally cumbersome to enforce the full influence-limiting policy on a cooperative spectrum sensing network with a large number of nodes, as the total number of subsets in $X$ with $n$ nodes is $\sum_{k=1}^{n} \binom{n}{k}$.

Therefore, we provide a more realistic strategy to reduce the complexity of the influence-limiting policy. We introduce a parameter $\eta, 0 < \eta \leq n$ in the influence-limiting policy and consider limiting the influence of any subset which has no more than $\eta$ elements. For example, $\eta = 1$ means that we perform influence-limiting policy only on individual nodes inside the network and $\eta = n$ means a full scale influence-limiting policy that evaluate on all possible combinations of $X_{\text{sub}}$. Through adjusting the value of $\eta$, we can balance the complexity of the influence-limiting policy and the effectiveness against a potential attack.

The step-by-step process of the influence-limiting policy can be seen from the Algorithm 2. The advantage of using influence-limiting policy is that we can add this enforcement as a parallel, non-invasive constraint to the primary/traditional decision method in the fusion center, because traditional spectrum sensing methods have a well-balanced goal to maximize the channel utilization. The influence-limiting policy is applied to bridge the gap between traditional defenses and the new adversarial machine learning based attacks, such as LEB attacks.

5.3 Experimental Validation

We then evaluate the performance of the proposed influence-limiting policy using the collected dataset. The experimental configurations are the same as those in Section 4. In this part of experiments, we adopt the trust value-based method [5] as the existing fusion center defense mechanism.

5.3.1 Measurement of the K-S Statistic

First, we examine the feasibility for the influence-limiting policy to use the K-S statistic of each node to set the threshold of $I(X_{\text{sub}})$. We illustrate the commutative probabilities of signal strengths of normal nodes (12 out of 20) before and after the LEB attack starts in Fig. 12 and those of malicious nodes (8 out of 20) in Fig. 13. From the figures, we can observe that when the LEB attack starts, malicious nodes tend to have a larger deviation of the statistical property from the training data than normal nodes. Thus, for any given $X_{\text{sub}},$
we can use $\sum_{j \in X_{\text{sub}}} d^2_{ij}$ in Eq. \((8)\) as an indication of their deviations from training data. The larger the deviation, the more limits we should place to curb its influence $I(\vert X_{\text{sub}} \vert)$ by reducing its threshold $\sigma(\vert X_{\text{sub}} \vert)$.

### 5.3.2 Impact of $c_1$ and $c_2$ for $\delta(\vert X_{\text{sub}} \vert)$

Next, we measure the impacts of parameters $c_1$ and $c_2$ with regard to the threshold $\delta(\vert X_{\text{sub}} \vert)$ in Eq\((8)\) on the defense performance of the influence-limiting policy.

It is obvious that the limitation caused by $\delta(\vert X_{\text{sub}} \vert)$ will lead to a performance cost for the fusion center, especially when there is no malicious node. The cost essentially depends on $c_1$ in Eq. \((8)\). According to the mathematical property of Sigmoid function, the smaller the value of $c_1$, the less limitation will be enforced; i.e., the cost when no malicious nodes are present will be smaller. On the other hand, however, a smaller $c_1$ makes the system suffer more damage when malicious nodes are indeed present. Similarly, a larger value of $c_2$ makes sure that a potentially malicious node will be penalized more but may also penalize legitimate nodes more when there is no attack. Therefore, we evaluate a wide range of choices of $c_1$ and $c_2$ to observe the balance between the performance without the attack and the defense effectiveness in the presence of the attack.

Case 1: No malicious node. In the case where no malicious node exists, all the 20 sensing nodes report the true sensed values to the fusion center. We illustrate the overall disruption ratio in 1000 timeslots with regard to influence-limiting policy of different values for $c_1$ and $c_2$ in Fig. \((14)\)(a). When evaluating one parameter, we fix the other parameter as the two parameters are independent from each other. We can observe from Fig. \((14)\)(a) that when no malicious node is present, the overall disruption ratio is increasing with the increase of both the values of $c_1$ and $c_2$, which is the slight cost of the influence-limiting policy.

Case 2: With LEB attacks. When LEB attacks are present, we evaluate the scenario where there exist 8 malicious nodes out of 20 nodes (i.e., $m = 8$ and $n = 20$). The relationship between different values of $c_1$ and $c_2$ and the overall disruption ratio is shown in Fig. \((14)\)(b). When the value of $c_1$ or $c_2$ is very small, the influence limitation is negligible. The defense performance is demonstrated when the values of the two parameters increased beyond 0.05 for $c_1$ and 0.5 for $c_2$.

For the influence-limiting policy to achieve its full potential of defense capability, we want to choose the parameter values of $c_1$ and $c_2$ that can decrease the overall disruption ratio dramatically when malicious nodes exist, while still maintains a slight cost when no malicious node is present. Combining the experimental results of the above two cases, we can observe that $c_1$ and $c_2$ can be chosen above 0.05 and 0.5, respectively.

### 5.3.3 Impact of $\eta$ on Defense

We also evaluate the defense performance under different values of $\eta$. The parameter $\eta$ balances the complexity and the defense performance of the influence-limiting policy. We adopt the settings of $c_1 = 0.6$, $c_2 = 0.08$ in this group of experiments. The overall disruption ratios under different numbers of malicious nodes with LEB attack are shown in Fig. \((15)\). The results demonstrate that when $\eta = 4$, the performance improvement is almost negligible for different cases of $m = 2, 4, 6, \text{and } 8$. 
TABLE 2: The overall disruption ratios with different numbers of malicious nodes $m$.

| LEB attacks | Influence-limiting policy | $m$ = 2 | $m$ = 4 | $m$ = 6 | $m$ = 8 |
|-------------|--------------------------|--------|--------|--------|--------|
| absent      | absent                   | 0.002  |        |        |        |
| present     | absent                   | 0.06   | 0.21   | 0.34   | 0.45   |
| absent      | present                  | 0.008  |        |        |        |
| present     | present                  | 0.013  | 0.05   | 0.082  | 0.095  |

5.3.4 Varying the Number of Malicious Nodes

Given the parameters $c_1$ and $c_2$, $\eta$ obtained, we conduct the comparisons of the influence-limiting policy-based defense with existing defenses in cooperative spectrum sensing.

We first compare the defense performance comparison of four cases in cooperative spectrum sensing: (i) No LEB attack and no influence-limiting policy in the fusion center; (ii) LEB attacks are present without the influence-limiting policy; (iii) No LEB attack while influence-limiting policy is enforced; (iv) Both LEB attacks and influence-limiting policy are present. We compared the performance in terms of the overall disruption ratio for $m = 2, 4, 6, 8$ in Table 2 which demonstrates that the cost of influence-limiting policy is very slight when no malicious node is present (with an overall disruption ratio of 0.008 compared to 0.002 when influence-limiting policy is absent). Compared with no influence-limiting policy is equipped, the average defense performance improvement of influence-limiting policy in terms of overall disruption ratio is around 78% when LEB attacks exist, which shows the effectiveness of the defense.

Besides the above comparisons, we also conduct the performance comparisons of influence-limiting policy-based defense with other existing defenses in terms of overall disruption ratio. We employ the same configurations for the parameters of influence-limiting policy, and set $m = 8$. The performance illustrated in Fig. 15 validates our proposed defense through dramatically decreasing the overall disruption ratio by around 80% in average.

The experiment results above demonstrate the robust performance of the proposed influence-limiting policy-based defense. When no influence-limiting policy is enforced, the LEB attacker tends to accumulate their decision flipping influence on the fusion center, further to compromise the decision process. The influence-limiting policy-based defense can effectively limit the attack capability of malicious nodes.

6 RELATED WORK

Cooperative Spectrum Sensing and Defense Cooperative spectrum sensing is an efficient way to detect the spectrum usages of TV white space channels. Defenses like statistics-based, machine learning-based, and trust-based methods have been widely studied. Statistics-based defense methods assign different statistics to the nodes. For example, an outlier is computed for each node in [8]; [9] obtains the weighted coefficients of each node; and [10] proposes a method through majority vote of neighboring nodes, a Bayesian method is discussed in [11]. Machine learning-based methods leverage machine learning techniques to classify legitimate and malicious data. For example, [6], [14] utilize supervised learning based on SVM to distinguish malicious nodes; [2] uses different KNN-based algorithms in an unsupervised way to detect malicious nodes; and [12], [13] discuss both supervised and unsupervised ways to defend the fusion center. A trust-based method [3], [5] maintains a trust value for each node, which will be used as a weight in the global decision process. Taking into account such a wide range of defenses, we present a stronger attack model based on the LEB framework, which aims to learn the defense and launch effective attacks.

Adversarial Machine Learning Adversarial machine learning focuses on learning under the existence of active adversaries [42]. The transferability [16] in machine learning models gives adversaries opportunities to learn and compromise the targeted models. Adversarial example generation methods, such as iterative methods [28], [29] and gradient-based methods [19], are efficient to create malicious data targeting a machine learning classifier under specific scenarios. Our proposed adversarial sensing data generation algorithm achieves the same performance as other existing methods, while reducing by 65% the computational cost on average.

7 CONCLUSION

In this paper, we propose a stronger attack model against cooperative spectrum sensing in real world, called the LEB framework, which offers an effective attack paradigm against cooperative spectrum sensing. The LEB attack is designed in a flexible and generic way and can be applied
to different attack scenarios. We use comprehensive experiments to show the severe impacts of LEB attacks brought to the cooperative spectrum sensing. Inspired from the gap between existing defenses and new adversarial learning based attacks, such as LEB attacks, we design a generic and non-invasive method named as influence-limiting policy, sided with existing defenses to counter the LEB-based and other similar attacks. The results demonstrate that influence-limiting policy can dramatically decrease the attack damages of malicious nodes.

Our research results shed light on the fact that the traditional duel of the attacks and defenses in cooperative spectrum sensing has to be re-visited in the presence of new learning-based attack models. We explored the partial model problem (i.e., the malicious attacker controls part of the input dimensions) in cooperative spectrum sensing. How the proposed solution to partial model problem in cooperative spectrum sensing can work in an encrypted scenario, in which the communications between the sensing nodes and fusion center are encrypted, will be our future work.

References

[1] L. Zhang, G. Ding, Q. Wu, Y. Zou, Z. Han, and J. Wang, “Byzantine attack and defense in cognitive radio networks: A survey,” Commun. Surveys Tuts., vol. 17, 2015.

[2] H. Li and Z. Han, “Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks,” IEEE Trans. Wireless Commun., vol. 9, 2010.

[3] R. Chen, J.-M. Park, and K. Bian, “Robust distributed spectrum sensing in cognitive radio networks,” in IEEE INFOCOM, 2008.

[4] P. Kalligineedi, M. Khabbazian, and V. K. Bhargava, “Malicious user detection in a cognitive radio cooperative sensing system,” IEEE Trans. Wireless Commun., vol. 9, 2010.

[5] A. S. Rawat, P. Anand, H. Chen, and P. K. Varshney, “Collaborative spectrum sensing in the presence of Byzantine attacks in cognitive radio networks,” IEEE Trans. Signal Process., vol. 59, 2011.

[6] O. Fatemieh, A. Farhadi, R. Chandra, and C. A. Gunter, “Using classification to protect the integrity of spectrum measurements in white space networks,” in NDSS, 2011.

[7] W. Wang, L. Chen, K. G. Shin, and L. Duan, “Secure cooperative spectrum sensing and access against intelligent malicious behaviors,” in IEEE INFOCOM, 2014.

[8] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. T. Hou, “Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks,” in IEEE INFOCOM, 2012.

[9] H. Chen, M. Zhou, L. Xie, and J. Li, “Cooperative spectrum sensing with M-ary quantized data in cognitive radio networks under SSSF attacks,” IEEE Trans. Wireless Commun., vol. 16, 2017.

[10] C. Chen, M. Song, C. Xin, and M. Alam, “A robust malicious user detection scheme in cooperative spectrum sensing,” in IEEE GLOBECOM, 2012.

[11] F. Penna, Y. Sun, L. Dooleck, and D. Cabric, “Detecting and countering statistical attacks in cooperative spectrum sensing,” IEEE Trans. Signal Process., vol. 60, 2012.

[12] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, “Machine learning techniques for cooperative spectrum sensing in cognitive radio networks,” IEEE J. Sel. Areas Commun., vol. 31, 2013.

[13] H. Wang and Y.-D. Yao, “Primary user boundary detection in cognitive radio networks: Estimated secondary user locations and impact of malicious secondary users,” IEEE Trans. Veh. Technol., vol. 67, 2018.

[14] S. Rajasegarar, C. Leckie, and M. Palaniswami, “Pattern based anomalous user detection in cognitive radio networks,” in IEEE ICASSP, 2015.

[15] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, “Towards the science of security and privacy in machine learning,” arXiv preprint arXiv:1611.03814, 2016.

[16] N. Papernot, P. McDaniel, and I. Goodfellow, “Transferability in machine learning: from phenomena to black-box attacks using adversarial samples,” arXiv preprint arXiv:1605.07277, 2016.
[42] J. Hayes and G. Danezis, “Machine learning as an adversarial service: Learning black-box adversarial examples,” arXiv preprint arXiv:1708.05207, 2017.