# Imbalanced Learning for Cooperative Spectrum Sensing in Cognitive Radio Networks

Lusi Li, He Jiang, and Haibo He
Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island
Kingston, RI 02881, USA
Email: {lli, hjiang, he}@ele.uri.edu

***Abstract*—We propose a novel cooperative spectrum sensing (CSS) framework for cognitive radio networks based on imbalanced learning techniques, which aims to resolve the skewed category distribution problems of signal data. For a radio channel shared by primary users (PUs) and secondary users (SUs), the signal data composed of energy vectors, in which each energy level is estimated by SU, can be used to detect the channel availability via a classifier. However, due to the nature of this application, the existing category-imbalance problem hinders the detection performance since the trained classifier has a better effect on the dominated category. To enhance the performance, sampling (e.g., oversampling, under-sampling, and combination) algorithms are employed to balance the training data set based on the imbalance degree metric of imbalance-ratio. The balanced training set then can be used to train classifiers with initial parameters, and the validation set can be utilized to tune as well as evaluate the classifiers. In the testing phase, the actual desired performance on unseen signal data can be determined based on the testing set, i.e., whether the channel is available or not. The performance of each sampling algorithm is measured in terms of receiver operating characteristic (ROC) curve and area under the ROC curve (AUC). The simulation results demonstrate the effectiveness of our proposed framework compared to traditional CSS methods.**

***Index Terms*—Cognitive radio, cooperative spectrum sensing, imbalanced learning, sampling algorithms, detection performance**

## I. Introduction

Cognitive radio (CR), as a promising technology in wireless communications, has emerged to enhance spectrum utilization such that the scarcity problem caused by limited radio spectrum can be alleviated [1]. Generally, a primary task of CR is to sense radio frequency (RF) environment and autonomously adjust transmission parameters, which provides new paths for spectrum access. In CR networks, primary users (PUs) have higher priorities to access the licensed spectrum, while secondary users (SUs) with lower priorities could access this spectrum only when the PUs are inactive. In order to maximize the performance of CR networks without interfering the PUs' usage, SUs need to have the ability of spectrum sensing, which is to identify and utilize the available spectrum that is not being used by any PU.

However, in practice, the issues of shadowing, multipath fading, and receiver uncertainty would result in the degradation of sensing performance. In this case, cooperative spectrum sensing (CSS) can be adopted, where SUs distributed in different locations cooperate to achieve higher sensing accuracy and reliability than individual SU does [2]. This cooperation can be implemented via a fusion center, combining shared sensing information and making decisions. There are two fusion schemes: hard fusion and soft fusion. With hard fusion scheme, only one-bit information is exchanged to determine whether the received energy exceeds a given threshold such as OR, AND, and the counting rules. With soft fusion scheme, specific energy levels estimated by SUs are transmitted, contributing to better decision-making for the fusion center [3].

Considering the learning, adaptive, and decision-making abilities [4]–[6], machine learning techniques enjoy the advantages compared to traditional CSS methods. Based on them, many approaches are proposed. In [7], a fusion center algorithm based on machine learning is designed to train the model per frame in real time and provide decision. Especially in [8], supervised learning methods (support vector machine and K-nearest neighbor) as well as unsupervised learning methods (K-means and Gaussian mixture model) are used to identify available and unavailable spectrum. Although the decision region could be discovered efficiently, most machine learning techniques are proposed to address classification problems based on an assumption of balanced category distributions. Whereas, it is not always true for a biased category distribution problem existing in many dynamic CR networks. For example, in the case of high active degree of PUs, the channel unavailable category (i.e., the majority) may be over-represented by a large number of energy vectors corresponding to the case that at least one PU is active, while the channel availability category (i.e., the minority) is under-represented by only a few energy vectors corresponding to the case that all PUs are inactive. The solutions for this problem using traditional learning methods bias the dominant category leading to poor detection performance of available spectrum and vice versa [9]–[11]. Thus the category-imbalance problem can be considered as a significant impediment to the success of CSS.

To overcome this impediment, we propose a novel CSS scheme based on binary imbalanced learning. In the context of CSS, an energy vector is considered as a feature vector, whose each component is an energy level estimated by each SU. What we need to do first is to measure the category-imbalance degree of the given energy vectors using imbalance-ratio (IR) [12], [13]. Then sampling approaches are used to achieve a balance between these binary categories. According

978-1-7281-0962-6/19/$31.00 ©2019 IEEE

to oversampling strategy, new minority energy vectors can be created by replicating original ones or generating synthetic ones. For the undersampling strategy, a subset of majority energy vectors can be removed to balance the vectors. Additionally, ensemble methods combined by over- and under-sampling strategies concurrently are used to generate minority vectors and remove majority vectors. The balanced vectors can be used to train classifiers and make decisions about channel availability.

The rest of this paper is organized as follows. Section II introduces the models for CSS. The imbalanced learning based CSS framework is presented in Section III. Experimental results and discussions are provided in Section IV followed by conclusions drawn in Section V.

## II. System Model

### A. Cognitive Radio Network Model

In order to compare the efficiency of our proposed framework with that in [8], we use the same CR network model. In a CR network, a frequency channel is shared by $N_s$ SUs and $N_p$ PUs. Let $c_{SU}^i$ indicate the coordinate of SU$^i$ in a 2-dimensional space, and $c_{PU}^j$ indicate the coordinate of PU$^j$ in the same space, where $i = 1, 2, \ldots, N_s$ and $j = 1, 2, \ldots, N_p$. In this paper, we assume that $N_p$ PUs can alternate between active and inactive states. $S = (S_1, S_2, \ldots, S_{N_p})^T$ denotes the states of PUs, in which $S_j$ is the state of PU$^j$ and $T$ is transpose operation. If PU$^j$ is transmitting a signal (i.e., PU$^j$ is active), we have $S_j = 1$. Similarly, if PU$^j$ is in inactive state, we have $S_j = 0$.

Given a state vector $S' = (S'_1, S'_2, \ldots, S'_{N_p})^T$ for all PUs, the probability of $S = S'$ can be represented as $P(S') = Pr[S = S']$. If $S_j = 0, \forall j$ (i.e., no PU is in the activate state), the channel can be considered as available for SUs to access. If $S_j = 1$ for some $j$ (i.e., at least one PU is in activate state), the channel is unavailable such that SUs have no right to access. The channel availability Y can be denoted by

$$Y = \begin{cases} 1, & \text{if } S_j = 0,\ \forall j \\ 0, & \text{if } S_j = 1,\ \exists j \end{cases} \tag{1}$$

For CSS, to detect the channel availability, each SU estimates the energy level of the channel and then reports it to the fusion center. According to the reported energy levels from all SUs, the fusion center decides whether the channel is available or not.

### B. Energy Vector Model

The energy detector is widely used due to its simplicity when the specific form is unknown in CR networks. At a time interval of $\tau$, the energy levels can be estimated by SUs. If we have the bandwidth of $\omega$, there would be $\omega\tau$ baseband signal samples captured by the energy detector. The signal samples are composed of the collection of that from thermal noise and all active PUs. $Z_i(k)$ indicates the $k$th signal sample captured by SU$^i$.

$$Z_i(k) = \sum_{j=1}^{N_p} h_{j,i} S_j X_j(k) + V_i(k), \quad k = 1, 2, \ldots, \omega\tau \tag{2}$$

where $h_{j,i}$ represents the channel gain from PU$^j$ to SU$^i$; $S_j$ denotes the state of PU$^j$; $X_j(k)$ is the signal transmitted by PU$^j$ during the $k$th detection;

$V_i(k)$ indicates the thermal noise at SU$^i$ during the $k$th detection. We assume that the transmission power of PU$^j$ is fixed to $d_j$.

$$d_j = \frac{1}{\tau} \sum_{k=1}^{\omega\tau} \mathrm{E}[|X_j(k)|^2] \tag{3}$$

The thermal noise spectral density is also fixed and given by $p_v = \mathrm{E}[|V_i(k)|^2]$. Thus, the estimated energy level through the energy detector of SU$^i$ after being normalized by $p_v$ can be obtained

$$f_i = \frac{2}{p_v} \sum_{k=1}^{\omega\tau} |Z_i(k)|^2, \quad i = 1, 2, \ldots, N_s \tag{4}$$

During each detection, the fusion center receives $N_s$ reported energy levels estimated by all SUs, and then it generates one energy vector for this detection, which is $W = (f_1, f_2, \ldots, f_{N_s})^T$. We assume that PUs and SUs are immobile, the consumer premise equipment, and the TV station in IEEE 802.22-based wireless regional area network. The multi-path fading and shadow fading components are quasi-static during the time of interest.

According to the central limit theorem, the distribution of the energy vector given $S = S'$ can be approximated by a Gaussian distribution if the value of $\omega\tau$ is large enough. The mean value can be shown as follows.

$$\mu_{f_i|S=S'} = E[f_i|S = S'] = 2\omega\tau + \frac{2\tau}{p_v} \sum_{j=1}^{N_p} \xi_{j,i}\, S_j\, d_j \tag{5}$$

where $\xi_{j,i} = (||c_{PU}^{N_p} - c_{SU}^{N_s}||^{-\varepsilon})\varphi_{j,i}\zeta_{j,i}$; $\varepsilon$ is the path-loss exponent; $\varphi_{j,i}$ represents shadow fading component from PU$^j$ to SU$^i$; $\zeta_{j,i}$ indicates multi-path fading component from PU$^j$ to SU$^i$. We can also have the variance.

$$\begin{aligned} \sigma^2_{f_i|S=S'} &= E[(f_i - \mu_{f_i|S=S'})^2|S = S'] \\ &= 4\omega\tau + \frac{8\tau}{p_v} \sum_{j=1}^{N_p} \xi_{j,i}\, S_j\, d_j \end{aligned} \tag{6}$$

Hence, the distribution of $\mathbf{W}$ can be considered as a multivariate Gaussian distribution. The mean vector and covariance matrix are respectively denoted by

$$\mu_{\mathbf{W}|S=S'} = (\mu_{f_1|S=S'}, \ldots, \mu_{f_{N_s}|S=S'})^T \tag{7}$$

$$C_{\mathbf{W}|S=S'} = \mathrm{diag}(\sigma^2_{f_1|S=S'}, \ldots, \sigma^2_{f_{N_s}|S=S'}) \tag{8}$$

where $C_{\mathbf{W}|S=S'}$ is a diagonal matrix, indicating that the components in $\mathbf{W}$ are independent.

## III. Imbalanced Learning based Cooperative Spectrum Sensing framework

### A. Overview of Proposed CSS Framework

Given a data set $\mathbf{W}$ (i.e., combination of energy vectors) and the corresponding channel availability $Y$, our proposed CSS

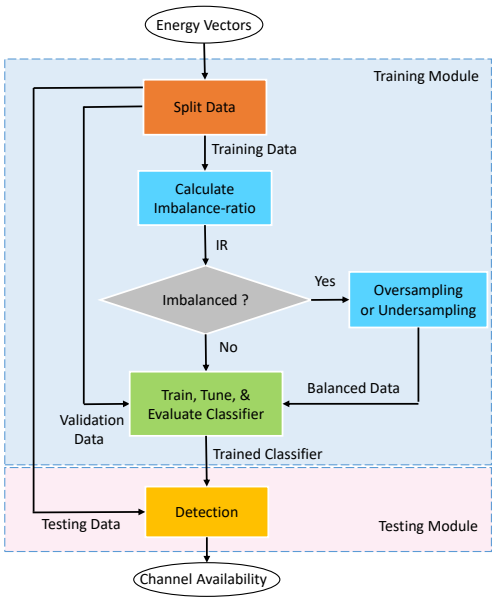



Fig. 1. The architecture of the proposed CSS framework

framework aims to efficiently detect whether the channel is available or not. With machine learning techniques, we need to build a classifier trained by labeled **W** and utilize it to project the input vectors to the channel availability $Y$.

Specifically, in this paper, there are two separated modules in our proposed framework: training module and testing module. In the first module, **W** can be split into three groups: 1) training set, which is used to train the classifier with initial learning parameters; 2) validation set, which is iteratively used to obtain the optimal parameters enabling the classifier to get best validation accuracy (i.e., cross-validation test); 3) testing data, which is used to get the test accuracy indicating the actual expected performance on unseen data. Therefore, the training energy vectors can be used to train a classifier, and the validation energy vectors contribute to tuning and evaluating the classifier. Finally, the testing energy vectors are classified by the trained classifier and we can have the channel availability results.

While the category-imbalance problem existing in energy vectors greatly hinders the detection performance of the channel availability. Thus, before training the classifier, what we should do is to determine the imbalance degree between the channel available category and the channel unavailable category in the training set (shown in Fig. 1). The imbalance-ratio (IR) is the most widely used metric of category imbalance for its simplicity. IR refers to the ratio of the number of energy vectors from the majority category to that from the minority category, indicating the category imbalance degree in size. The category with a larger number of vectors can be considered as the majority category, and that with a smaller number of vectors can be viewed as the minority category. IR = 1 when the training data is balanced. It can be known that the larger the value of IR, the more imbalanced the training data is. Consequently, the trained classifier would have a good effect on the detection of the majority category and have poor detection performance of the minority category.

When we get IR $> 1$, the training data is imbalanced. Then we can use oversampling methods to generate minority samples or under-sampling methods to remove majority samples or even the combination of over- and under-sampling methods such that the balanced data can be obtained. After balancing the data, we could use them for training the classifier. If IR = 1, we can directly build the model. The validation data as a data set held back from training the classifier can be used to estimate its performance and tune its parameters before utilizing. In the second module, the detection performance of trained classifier can be tested by the testing data.

In a word, the training module shown in the blue part of Fig. 1 attempts to balance the training energy vectors via sampling methods when given data are imbalanced. Then it is used to train a classifier based on the balanced energy vectors, and offers the trained classifier to classification module shown in pink part. In the CR network, we can activate the training module at first deployment and when RF environment changes. Moreover, it would be activated periodically to adapt to the variety of environment, or even run in the background when we operate the classification module under normal use. Based on this mechanism, the time required to train the model is not a big issue.

For clear presentation, some notations are described here. Given a data set **W** with M energy level samples, **W** can be split into the training set $W_r = \{(w_r^i, y_r^i)\}, i = 1, 2, \ldots, M_r$, the validation set $W_v = \{(w_v^j, y_v^j)\}, j = 1, 2, \ldots, M_v$, and the testing set $W_e = \{(w_e^l, y_e^l)\}, l = 1, 2, \ldots, M_e$, where each $w \in \mathbf{W}$ is a sample in the $N_s$-dimensional feature space $W = \{f_1, f_2, \ldots, f_{N_s}\}$, and $y \in Y = \{0, 1\}$ is a category identity label corresponding to sample $w$. Moreover, the training subsets $W_r^+, W_r^- \subset W_r$ are defined, respectively, in which $W_r^+$ is the training set of majority category samples in $W_r$ and $W_r^-$ is the training set of minority category samples in $W_r$ such that $W_r^+ \cup W_r^- = \{W_r\}$ and $W_r^+ \cap W_r^- = \{\Phi\}$. The generated minority set can be denoted as $G_-$ and the removed majority set can be denoted as $G_+$.

### B. Sampling Methods

The traditional sampling methods include random oversampling (ROS) and random under-sampling (RUS) methods. To achieve a balance, the former method replicates the original minority samples by $|G_-|$, where generally $0 \leq |G_-| \leq |W_r^+| - |W_r^-|$. It provides a scheme for altering the imbalance degree of data to an expected level. The latter method is to randomly select $|G_+|$ original majority samples and remove them, in which $0 \leq |G_+| \leq |W_r^+| - |W_r^-|$. However, this strategy by introducing its own set not by changing its distribution would hinder learning. Therefore, a large number of methods are proposed. In this section, some imbalanced learning algorithms are presented, i.e., synthetic minority oversampling technique (SMOTE) [14], adaptive synthetic (ADASYN) sampling approach [15], EasyEnsemble

[16], BalanceCascade methods [16], SMOTEENN [17], and SMOTETomek [18] for the proposed CSS framework.

*1) SMOTE:* The SMOTE, as one of the widely used synthetic-based sampling approaches, has achieved some success in many applications. It generates synthetic samples based on the distances between minority samples in the feature space. Specifically, the $K$-nearest neighbors of each sample $w_r^i$ in subset $W_r^-$ can be obtained for a given $K$, which have the top $K$ smallest euclidian distances with $w_r^i$ in the $N_s$-dimensional feature space. Then one of the $K$-nearest neighbors can be randomly selected, multiplied by a random vector with each element from 0 to 1, and then added to $w_r^i$. Thus a synthetic minority sample $w_r^{'}$ can be created using the following formula:

$$w_r^{'} = w_r^i + (\hat{w_r^i} - w_r^i). * \lambda \tag{9}$$

where $\hat{w_r^i} \in W_r^-$ is one of the $K$-nearest neighbors of $w_r^i$, and $\lambda$ is a random vector indicating the feature vector difference. Hence, $w_r^{'}$ is definitely a minority sample along the line between $w_r^i$ and $\hat{w_r^i}$. Finally, the generated set $G_-$ are composed of $w_r^{'}s$.

*2) ADASYN:* Compared to SMOTE, ADASYN can measure the distributions of samples, i.e., determine the difficulty levels for learning minority samples. Then different numbers of synthetic samples can be adaptively generated around those identified minority ones with high levels. First of all, the required amount of synthetic samples for the minority category should be computed by $n = (|W_r^+| - |W_r^-|) \times \delta$, where $|W_r^+|$ indicates the number of samples in subset $W_r^+$; $|W_r^-|$ represents the number of samples in subset $W_r^-$; $\delta$ is a random number between 0 and 1, which represents the expected balance degree after generation procedure of synthetic samples. In the second step, unlike SMOTE, the $K$-nearest neighbors of each sample $w_r^i$ in the entire training set $W_r$ can be found using the euclidean distance metric, which are used to get the ratio $\gamma_i$

$$\gamma_i = \frac{\Theta_i}{KC}, \quad i = 1, 2, \ldots, |W_r^-| \tag{10}$$

where $\Theta_i$ indicates the number of majority samples in the $K$-nearest neighbors of $w_r^i$, which belongs to the minority category; $\sum \gamma_i = 1$, since $C$ as a normalization constant allows $\gamma_i$ in [0,1]. Next, for each minority sample $w_r^i$, the amount of synthetic samples that require to create can be given by $g_r^i = n \times \gamma_i$. Thus, we can create $g_r^i$ synthetic samples using the Equation (12) to obtain $G_-$. The ADASYN method mainly focuses on the metric of density distribution $\gamma$ to determine how many synthetic samples can be generated for each sample in minority category, enabling the weights of them to adaptively alter.

*3) EasyEnsemble and BalanceCascade:* Random undersampling method may raise the problem of information loss, since the removed majority samples are randomly selected. To overcome this deficiency, EasyEnsemble and BalanceCascade approaches are proposed. EasyEnsemble uses an ensemble learning scheme: first several subsets with $|W_r^-|$ (not $|W_r^+|$) samples in majority category are produced according to the rule of independent and random sampling, and then they are combined with the entire minority samples, which are considered as the training set of multiple classifiers. To make a final decision, all the trained classifiers are combined by

$$H(w_r) = sgn(\sum_{a=1}^{A} \sum_{b=1}^{B_a} \pi_{a,b} h_{a,b}(w_r) - \sum_{a=1}^{A} \theta_a) \tag{11}$$

where $H(w_r)$ is the ensemble classifier; $A$ indicates the number of produced subsets; $B_a$ is the iteration number to train weak classifiers $h_{a,b}(w_r)$; $\pi_{a,b}$ are the corresponding weights of $h_{a,b}(w_r)$; $\theta_a$ represents the threshold of the ensemble.

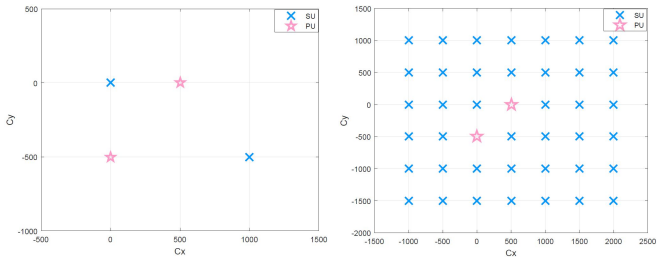
Unlike EasyEnsemble, which can be viewed as an unsupervised learning method, BalanceCascade explores the importance of each majority sample. Specially, we get trained classifier $H_1$ at the first iteration. Then at the second iteration, the sample $w_r^i \in W_r^+$ can be considered as a redundant one in $W_r^+$ if $H_1$ classifies it correctly. Hence, $w_r^i$ would be removed from the majority category. This procedure will to be continued until the termination conditions are reached. The output of BalanceCascade is the same with EasyEnsemble but has a different training strategy.

*4) SMOTETomek and SMOTEENN:* Tomek link, as one of the classic data cleaning approaches, can be applied to alleviate the overlapping problem between categories, especially for the majority category. It describes a special relationship between two samples from different categories. Given any pair $w_r^i \in W_r^+$ and $w_r^j \in W_r^-$, $d(w_r^i, w_r^j)$ indicates the distance between $w_r^i$ and $w_r^j$. Then $(w_r^i, w_r^j)$ can be considered as a Tomek link if $w_r^q \in W_r$ has no existence so that $d(w_r^i, w_r^q) < d(w_r^i, w_r^j)$ or $d(w_r^j, w_r^q) < d(w_r^i, w_r^j)$. Hence, it can be known that either $w_r^i$ or $w_r^j$ is noise or both of them are located near the border. According to the principle of Tomek link, the overlapping problem can be mitigated by removing all links until all pairs belong to the same category. Based on this, SMOTETomek combines the idea of SMOTE with Tomek to perform oversampling and undersampling concurrently. In addition, SMOTE also is combined with edited nearest neighbor (ENN) rule to form SMOTEENN method, which removes samples that distribute in different category with two of their three nearest neighbors.

## IV. Simulations

### A. Setup

In this simulation of cooperative spectrum sensing, we first present a 2-dimensional scenario, where there are 2 PUs and 2 SUs in a 5-by-5 grid topology within a 2000m $\times$ 1500m area shown in Fig. 2. (a). Then in the next multi-dimensional scenario, we fix the locations of 2 PUs with (0m, -500m) and (500m, 0m) in the first scenario and increase the number of SUs to 40 in a 4000m $\times$ 3500m area shown in Fig. 2. (b). The key parameters are set as follows: the sensing time interval $\tau = 100\mu s$, the bandwidth $\omega = 10MHz$, the transmission power of each PU $d = 200mW$, the path-loss exponent $\varepsilon = 4$, the thermal noise spectral density $p_v = -174dBm$, fixed shadow fading component $\varphi_{j,i} = 1$, fixed multi-path




(a) topology with 2 SUs (b) topology with 40 SUs

Fig. 2. The CR network topology applied for simulations

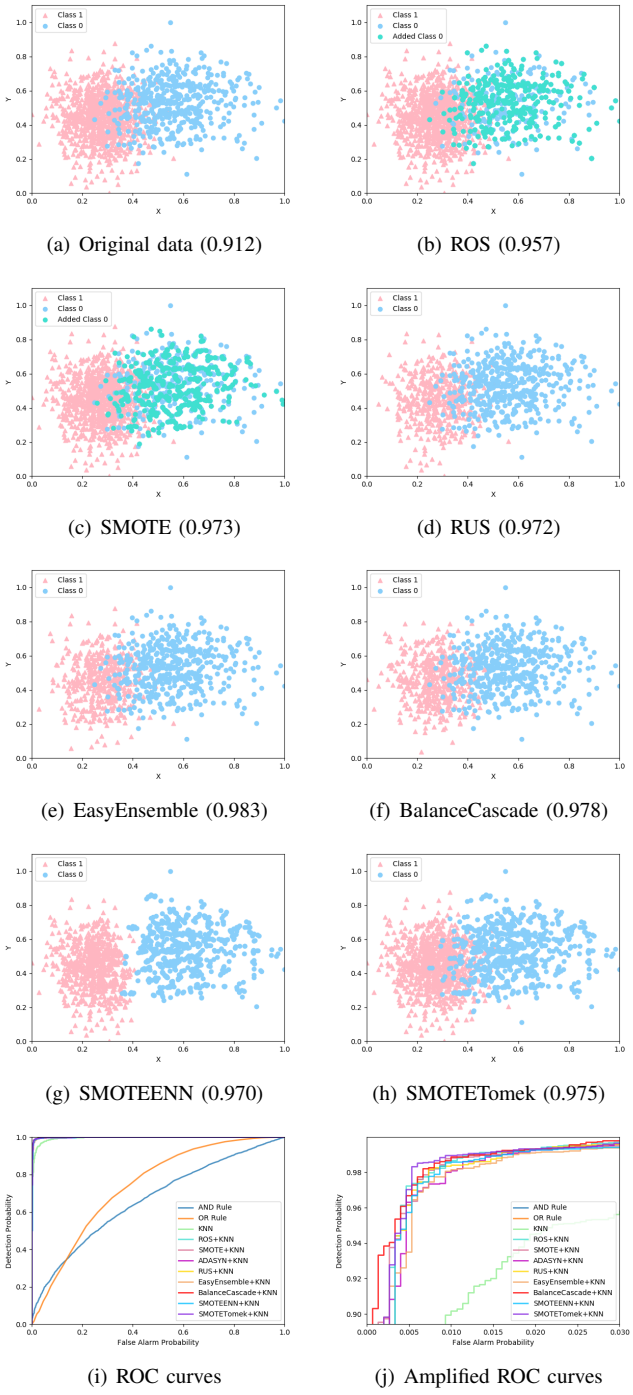
fading component $\zeta_{j,i} = 1$. The probabilities that either one or both PUs are in the activate state result in different category-imbalanced degrees. The states of 2 PUs are also independent. In this paper, we assume the channel available category is the majority and the channel unavailable category is the minority. The reverse is also true.

Specially, the detection performance of our proposed framework is compared with two hard fusion methods (i.e., AND rule and OR rule), and several imbalance leaning methods, including ROS, SMOTE, ADASYN, RUS, EasyEnsemble, BalanceCascade, SMOTEENN, and SMOTETomek. With oversampling techniques, the minority samples are oversampled until their number reaches the number of the majority samples. With undersampling techniques, the majority samples are undersampled until their number is decreased to the number of the minority samples. Additionally, we select two common used base classifiers including K-nearest neighbors (KNN) and decision tree, in which K is set to 5. All the other parameters of these methods are set as default.

Imbalanced learning techniques aim to enhance the detection performance for the minority category. In this paper, two typical metrics for imbalanced learning are used to measure the performance: AUC and ROC curve. Our proposed framework and all the compared methods are implemented by Python 3.6 in a 64-bit computer with an Intel i7-6700 CPU.

### *B. 2-Dimensional Scenario*

In this scenario, the 2 PUs in Fig. 2 (a) are activated based on the probability $p_s((0,0)^T) = 0.67$, $p_s((0,1)^T) = 0.11$, $p_s((1,0)^T) = 0.11$, and $p_s((1,1)^T) = 0.11$. 1200 energy samples are generated for PUs in $S = (0,0)^T$, 200 ones for PUs in $S = (0,1)^T$, 200 ones for PUs in $S = (1,0)^T$, and 200 ones for PUs in $S = (1,1)^T$. The IR for the entire energy data can be calculated: IR = 1200/600 = 2, where the channel available category as the majority has 1200 samples, and the channel unavailable category as the minority involves 600=200+200+200 samples. For simplicity, the 1800 energy samples are randomly divided into the training set with 1200 samples, the validation set with 300 samples, and the testing set with 300 samples. To ensure consistency with the entire data set, IR value in the training set can also be set 2. The original training set and all the final synthetic energy vectors are visualized with 2-dimensional scatter plots to compare their performance as shown in Fig. 3. The pink and




(a) Original data (0.912) (b) ROS (0.957)

(c) SMOTE (0.973) (d) RUS (0.972)

(e) EasyEnsemble (0.983) (f) BalanceCascade (0.978)

(g) SMOTEENN (0.970) (h) SMOTETomek (0.975)

(i) ROC curves (j) Amplified ROC curves

Fig. 3. Scatter plots of the 2-D original energy data set in (a) and synthetic energy data using ROS, SMOTE, RUS, EasyEnsemble, BalanceCascade, SMOTEENN, and SMOTETomek in (b)-(h). ROC curves and their amplified curves in (i) and (j).

blue dots symbolize majority samples and minority samples, respectively from class 1 and 0. Fig. 3 (a) shows the scatter plot of original energy data, which is directly used to train KNN. The green dots denote the samples generated by ROS in Fig. 3 (b) and SMOTE in Fig. 3 (c). Then the obtained training set can be as the input of the base classifier of KNN. The performance is measured by AUC. From the AUC values, it can be known that using sampling methods could achieve

better detection performance than no use.

TABLE I
AVERAGE AUC VALUES FOR TRAINING SETS WITH MULTI-D AND DIFFERENT IRS BY VARIOUS METHODS

| Methods | | IR of Training Set | | | | |
|---|---|---|---|---|---|---|
| | | ***2*** | ***4*** | ***6*** | ***8*** | ***10*** |
| **Hard Fusion** | AND Rule | 0.667 | 0.674 | 0.638 | 0.642 | 0.680 |
| | OR Rule | 0.666 | 0.650 | 0.668 | 0.704 | 0.675 |
| **Soft Fusion** (KNN) | Baseline | 0.908 | 0.913 | 0.898 | 0.897 | 0.864 |
| | ROS | 0.956 | 0.930 | 0.916 | 0.917 | 0.875 |
| | SMOTE | 0.972 | **0.978** | 0.972 | 0.976 | 0.962 |
| | ADASYN | 0.968 | 0.963 | 0.970 | 0.978 | 0.960 |
| | RUS | 0.965 | 0.967 | 0.975 | 0.973 | 0.960 |
| | EasyEnsemble | 0.970 | 0.974 | **0.980** | **0.979** | 0.964 |
| | BalanceCascade | 0.973 | 0.966 | 0.975 | 0.972 | **0.974** |
| | SMOTEENN | 0.972 | 0.970 | 0.975 | 0.971 | 0.959 |
| | SMOTETomek | **0.975** | 0.968 | 0.978 | 0.971 | 0.961 |
| **Soft Fusion** (Decision Tree) | Baseline | 0.913 | 0.930 | 0.906 | 0.932 | 0.912 |
| | ROS | 0.954 | 0.937 | 0.923 | 0.948 | 0.926 |
| | SMOTE | 0.967 | 0.963 | **0.968** | **0.956** | 0.934 |
| | ADASYN | 0.963 | **0.965** | 0.959 | 0.955 | 0.936 |
| | RUS | 0.963 | 0.948 | 0.951 | 0.952 | 0.950 |
| | EasyEnsemble | 0.962 | 0.958 | 0.958 | 0.938 | **0.965** |
| | BalanceCascade | **0.969** | 0.956 | 0.950 | 0.953 | 0.962 |
| | SMOTEENN | 0.967 | 0.946 | 0.956 | 0.948 | 0.943 |
| | SMOTETomek | 0.964 | 0.947 | 0.951 | 0.941 | 0.955 |

### C. Multi-Dimensional Scenario

In this scenario, we add the number of SUs to 40 for estimating the energy vectors. The 2 PUs in Fig. 2 (b) are activated based on varying probabilities. They also have 4 states. For total of 12000 energy vectors that need to produce, to observe the detection performance of these sampling methods, we set different IR values for the entire energy vectors in line with the training set. For an example, when IR = 4, there would be 9600 energy samples for PUs in in $S = (0,0)^T$, 800 ones for PUs in $S = (0,1)^T$, 800 ones for PUs in $S = (1,0)^T$, and 800 ones for PUs in $S = (1,1)^T$. We perform 6-fold cross validation to get the average AUC values. In cross validation procedure, each fold can be as the testing set; one of the rest 5 folds can be as the validation set; 4 folds are training set with corresponding given IR. The average AUC values for different sampling methods with KNN and decision tree classifiers are shown in Table I. From the results, all machine learning methods perform better than the traditional CSS methods. Moreover, classifiers with sampling methods outperform the others. To further demonstrate the effectiveness of our proposed framework, we plot the average ROC curve of each detection method using the two base classifiers for IR=10 in Fig. 3. (i). Fig. 3. (j) represents the amplified of (i) for clearly observing the performance of each sampling method. It can be noticed that all sampling methods combined with base classifiers achieve better detection performance than the original classifiers and BalanceCascade performs best.

## V. CONCLUSION

In this paper, we present a novel imbalanced learning based CSS framework, aiming to solve the category-imbalanced problem in CR networks. After splitting the given energy vectors estimated by SUs into three groups, we first determine the imbalance degree of the training set with imbalance-ratio, and then balanced it using various sampling methods. The balanced set can be applied to train the classifier and the validation set can be used to tune the classifier. Finally, we obtain the channel availability of the testing set by the trained classifiers. Simulation results show the effectiveness of our proposed CSS framework.

## VI. ACKNOWLEDGMENT

This work was supported by the National Science Foundation under ECCS 1731672.

## REFERENCES

[1] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE journal on selected areas in communications*, vol. 23, no. 2, pp. 201–220, 2005.

[2] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical communication*, vol. 4, no. 1, pp. 40–62, 2011.

[3] B. Khalfi, B. Hamdaoui, and M. Guizani, "Airmap: Scalable spectrum occupancy recovery using local low-rank matrixapproximation," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 206–212.

[4] Z. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, and H. He, "Robot-assisted pedestrian regulation based on deep reinforcement learning," *IEEE transactions on cybernetics*, 2018.

[5] S. Li, L. Li, J. Yan, and H. He, "SDE: A novel clustering framework based on sparsity-density entropy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1575–1587, 2018.

[6] L. Li, H. He, J. Li, and W. Li, "EDOS: Entropy difference-based oversampling approach for imbalanced learning," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[7] A. M. Mikaeil, B. Guo, and Z. Wang, "Machine learning to data fusion approach for cooperative spectrum sensing," in *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. IEEE, 2014, pp. 429–434.

[8] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on selected areas in communications*, vol. 31, no. 11, pp. 2209–2221, 2013.

[9] J. Li, H. He, and L. Li, "CGAN-MBL for reliability assessment with imbalanced transmission gear data," *IEEE Transactions on Instrumentation and Measurement*, 2018.

[10] Z. Wan, H. He, and B. Tang, "A generative model for sparse hyperparameter determination," *IEEE Transactions on Big Data*, vol. 4, no. 1, pp. 2–10, 2018.

[11] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge & Data Engineering*, no. 9, pp. 1263–1284, 2008.

[12] B. Tang and H. He, "Gir-based ensemble sampling approaches for imbalanced learning," *Pattern Recognition*, vol. 71, pp. 306–319, 2017.

[13] L. Li, H. He, and J. Li, "Entropy-based sampling approaches for multi-class imbalanced problems," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[15] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks*. IEEE, 2008, pp. 1322–1328.

[16] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2009.

[17] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[18] G. E. Batista, A. L. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: a case study." in *WOB*, 2003, pp. 10–18.