

SVM and Decision Stumps Based Hybrid AdaBoost Classification Algorithm for Cognitive Radios

Siji Chen, Bin Shen, Xin Wang, and Hebiao Wu

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications(CQUPT)

Key Laboratory of Mobile Communication Technology, CQUPT

Chongqing 400-065, P. R. China

Email: cqchensj@outlook.com; shenbin@cqupt.edu.cn; 1306182550@qq.com; 1137544348@qq.com

Abstract—In this paper machine learning techniques based cooperative spectrum sensing (CSS) algorithms are investigated for cognitive radio networks (CRN). A novel support vector machine (SVM) and decision stumps based AdaBoost classification algorithm is proposed for pattern classification of the primary user's behavior in the network. Conventionally, AdaBoost algorithm combines multiple sub-classifiers and produces a strong weight based on their own weights in classification. Taking into account the fact that SVM and decision stump serve as relatively strong and weak classifiers respectively, the proposed algorithm employs SVM as the first-stage classifier and decision stump as the second-stage classifiers to eventually determine the class that the spectrum energy vector belongs to. It is verified in simulations that the proposed algorithm is capable of achieving higher detection probability than the conventional machine learning algorithms.

Index Terms—Support vector machine, decision stump, AdaBoost, energy vector, cooperative spectrum sensing, cognitive radio network (CRN).

I. INTRODUCTION

Cognitive radio (CR) enables dynamic spectrum access of the licensed spectrum in an opportunistic manner which may tremendously ease the increasing pressure on current spectrum resource allocation regimes [1]. The spectrum acquisition capability of CR systems generally relies on the spectrum sensing technologies they adopt to identify the spectrum status [2]. The secondary users (SU) have to discern the status of the licensed spectrum and access it only if they could assure themselves that the licensed spectrum is not temporarily occupied by the primary users (PU). Among the various spectrum sensing methods, energy detection (ED) is a commonly used technique that requires no priori knowledge of the PU signal [3]. However, the energy detectors are vulnerable and could not perform well under low SNR conditions, especially in multipath fading and shadowing environments.

In order to tackle the problem aroused in single user based spectrum sensing, cooperative spectrum sensing (CSS) are widely addressed, where multiple SUs collaborate to make a global decision on the licensed spectrum status. During each sensing interval, all SUs will report their spectrum observations or decision results to the fusion center (FC) and subsequently the FC makes a final decision based on its predefined decision strategy or fusion criterion [4]. In [5],

machine learning techniques, for instance the support vector machine (SVM), are studied, with an aim to find a linearly separable hyperplane with the help of support vectors by maximizing the margin of the classifier while minimizing the sum of classification errors [6]. In [7] [8], adaptive boosting (AdaBoost) algorithm is introduced, which is a model composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall prediction.

We propose in this paper a hybrid combining AdaBoost classification algorithm which uses SVM and decision stumps as sub-classifiers for cooperative spectrum sensing in cognitive radio networks. Taking into account the fact that SVM and decision stump serve as relatively strong and weak classifiers respectively, the proposed algorithm employs SVM as the first-stage classifier and decision stump as the second-stage classifiers to eventually determine the class that the spectrum energy vector belongs to. In a cognitive radio network (CRN) with a single PU and N SUs, the proposed hybrid AdaBoost classification algorithm could yield higher detection probability than the SVM classification algorithm and the decision stumps based AdaBoost classification algorithm as well. When compared with the traditional CSS schemes, for example, the AND and OR rule based hard decision fusion algorithms, the proposed algorithm outperforms them too. All these verifications prove that the proposed algorithm could be utilized in practice as a new solution for CSS.

This paper is organized as follows. In Section II, we present the system model and some assumptions. Section III introduces the hybrid AdaBoost classification algorithm in detail. In section IV, we present the performance evaluation results for the proposed hybrid combining classification algorithm. Finally, section V concludes the paper.

II. SYSTEM MODEL

We consider a CR network consists of a PU transmitter and N SU receivers, each of which is indexed by $n = 1, 2, \dots, N$. We assume the probability of the PU being active is P_{ON} and we have two hypotheses which are H_0 and H_1 with probability $(1 - P_{ON})$ and P_{ON} , respectively.

The received signal at the n -th SU is given as

$$y_n(i) = \begin{cases} r_n(i), & H_0 \\ h_n x(i) + r_n(i), & H_1. \end{cases} \quad (1)$$

where hypothesis H_0 indicates that the PU is inactive and the corresponding channel is available for SUs, while hypothesis H_1 means that the PU is active and the channel is unavailable for SUs; h_n denotes the channel gain from the PU to SU_n , $x(i)$ the PU's transmitting signal, and $r_n(i)$ the additive white gaussian noise (AWGN) with zero-mean and variance $\eta = E[|r_n(i)|^2] = \sigma^2$. If we denote the time duration of each spectrum sensing interval at each SU as τ and the spectrum bandwidth as ω , each SU takes $\omega\tau$ samples of its received signal during the sensing duration τ .

The noise variance normalized energy statistic of SU_n in the first CSS phase can be denoted by Y_n :

$$Y_n = \frac{1}{\eta} \sum_{i=1}^{\omega\tau} |y_n(i)|^2 \quad (2)$$

In the second CSS phase, all SUs report their spectrum observations Y_n to the FC, which generates the energy vector as:

$$\mathbf{Y} = [Y_1, Y_2, \dots, Y_N]^T \quad (3)$$

The power attenuations between the PU and the SUs depend on the channel coefficient h_n in that $g_n = |h_n|^2$. The power attenuation coefficient g_n is hence expressed as:

$$g_n = PL(\|c_{PU} - c_n^{SU}\|) \cdot \psi_n \cdot v_n \quad (4)$$

where $\|\cdot\|$ is the Euclidean distance, $PL(d) = d^{-\alpha}$ the pathloss component for the relative distance d with the pathloss exponent α , ψ_n the shadow fading component, and v_n the multipath fading component.

In machine learning, we need to obtain enough training energy vectors as our training samples to train a classifier [8]. Let $\mathbf{x}^{(l)}$ denote the l -th training energy vector and $y^{(l)}$ the channel availability class corresponding to $\mathbf{x}^{(l)}$. Thus, the training energy vectors are $\bar{\mathbf{x}} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(L)}\}$ and the channel availability flag (or label) is represented by $\bar{y} = \{c^{(1)}, c^{(2)}, \dots, c^{(L)}\}$, where L is the number of training samples. After the classifier is successfully trained over the given training samples, it is able to classify the input test energy vector and we use c^* to denote its corresponding channel availability. In addition, the set of test energy vectors can be denoted as $\mathbf{x}^* = \{\mathbf{x}^{*(1)}, \mathbf{x}^{*(2)}, \dots, \mathbf{x}^{*(M)}\}$ and the set of corresponding channel availability is $c^* = \{c^{*(1)}, c^{*(2)}, \dots, c^{*(M)}\}$, where M is the number of test samples. Suppose $\hat{c}^{(m)}$ denotes the m -th test sample's channel availability label decided by the classifier which is either channel available label (i.e., $\hat{c}^{(m)} = -1$) or channel unavailable label (i.e., $\hat{c}^{(m)} = 1$). If the test energy vector is classified into the channel available class, it means that there is no PU in the active state and the channel is available for the CR network to access. Therefore, the channel availability is correctly determined in the case that $\hat{c}^{(m)} = c^{*(m)}$,

while miss detection (or false alarm) occurs in the case that $\hat{c}^{(m)} = -1$ and $c^{*(m)} = 1$ (or $\hat{c}^{(m)} = 1$ and $c^{*(m)} = -1$). Thus, the prediction error (P_e) is defined as the probability of mis-predicting the presence of PU:

$$P_e = \frac{\sum_{m=1}^M I(\hat{c}^{(m)} \neq c^{*(m)})}{M} \quad (5)$$

where $I(\cdot)$ is the indicator function.

III. HYBRID ADABOOST CLASSIFICATION ALGORITHM

In our work, the typical machine learning methods of decision stump and SVM are first considered. A decision stump is a one-dimension decision tree, which uses only a single attribute for splitting and makes only one judgment on each attribute (i.e., one of the sample's dimensions). Decision stumps are often used as sub-classifiers in ensemble methods.

The SVM provides a binary model in machine learning which strives to find a linearly separable hyperplane with the help of support vectors (i.e., energy vectors that lie closest to the decision surface) by maximizing the margin of the classifier while minimizing the sum of classification errors [9]. As shown in Fig. 1, the learning strategy of SVM is to maximize the margin and its learning goal is to find a hyperplane in the N -dimensional samples space. The hyperplane equation can be expressed as

$$\omega^T \mathbf{x}^{(l)} + b = 0 \quad (6)$$

where ω is the weighting vector and b is the bias. Based on ω , we need to minimize the vector norm of ω so as to maximize the category margin, and hence the objective function is

$$\begin{aligned} \text{Min} \quad & \frac{1}{2} \|\omega\|^2 \\ \text{s.t.} \quad & c^{(l)}(\omega^T \mathbf{x}^{(l)} + b) \geq 1 \end{aligned} \quad (7)$$

Therefore, the SVM should satisfy the following condition for all $l = 1, 2, \dots, L$:

$$c^{(l)} = \begin{cases} 1, & \text{if } \omega^T \mathbf{x}^{(l)} + b \geq 1 \\ -1, & \text{if } \omega^T \mathbf{x}^{(l)} + b \leq -1 \end{cases} \quad (8)$$

In practice, when a test energy vector $\mathbf{x}^{*(m)}$ is fed into the SVM model, the SVM can determine which class it belongs to through the following rules:

$$\hat{c}^{(m)} = \begin{cases} 1, & \text{if } \omega^T \mathbf{x}^{*(m)} + b \geq 1 \\ -1, & \text{if } \omega^T \mathbf{x}^{*(m)} + b \leq -1 \end{cases} \quad (9)$$

However, practically for most of the time, the samples are not linearly separable. For this case, the hyperplane satisfying such conditions does not exist at all. Then, we need to find a fixed nonlinear mapping ϕ to map the non-linear samples into a new feature space and use a linear SVM in the feature space

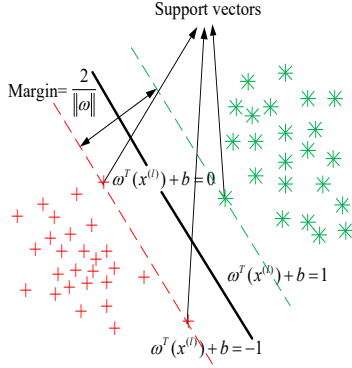


Fig. 1: SVM model

[10]. Hence, the nonlinear SVM should satisfy the following condition for all $l = 1, 2, \dots, L$:

$$c^{(l)} = \begin{cases} 1, & \text{if } \omega^T \phi(\mathbf{x}^{(l)}) + b \geq 1 \\ -1, & \text{if } \omega^T \phi(\mathbf{x}^{(l)}) + b \leq -1 \end{cases} \quad (10)$$

where $\phi(\mathbf{x})$ is the feature map function.

And the decision rule for nonlinearly SVM is given as:

$$\hat{c}^{(m)} = \begin{cases} 1, & \text{if } \omega^T \phi(\mathbf{x}^{*(m)}) + b \geq 1 \\ -1, & \text{if } \omega^T \phi(\mathbf{x}^{*(m)}) + b \leq -1 \end{cases} \quad (11)$$

Although the training energy vectors have been mapped into a higher dimensional feature space, practically we cannot achieve a perfect linearly separable hyperplane that satisfies the condition in (11) for each $\mathbf{x}^{(l)}$. Hence, we rewrite the optimization problem as a convex optimization problem as follows:

$$\begin{aligned} \text{Min} \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{l=1}^L \xi^{(l)} \\ \text{s.t.} \quad & c^{(l)}(\omega^T \mathbf{x}^{(l)} + b) \geq 1 - \xi^{(l)}, l = 1, 2, \dots, L \\ & \xi^{(l)} \geq 0, l = 1, 2, \dots, L \end{aligned} \quad (12)$$

where C is a soft margin constant, for which a larger C means the assignment of a higher penalty to errors, and $\xi^{(l)}$ is a slack variable. Radial basis function (rbf) kernel is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in SVM classification to map a feature space to a higher dimension.

Fig. 2 and Fig. 3 show the training samples classified by SVM-rbf. We can notice that the decision surface divides the energy vectors in each class as clearly as possible, which leads to improved detection performance. Also, decision surface can separate energy vectors more accurately as the SNR increases. This decision surface takes a complex form, which cannot be described by any other existing CSS technique.

In addition to decision stump and SVM, AdaBoost (Adaptive Boosting) algorithm is an ensemble learning algorithm composed of plenty of sub-classifiers to overcome the

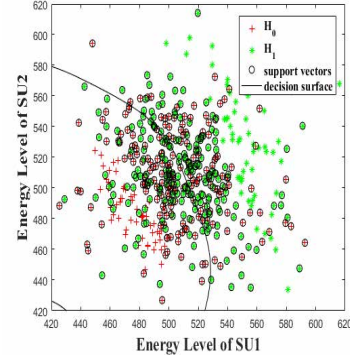


Fig. 2: Scatter plot of energy vectors classified by SVM when SNR=-1dB

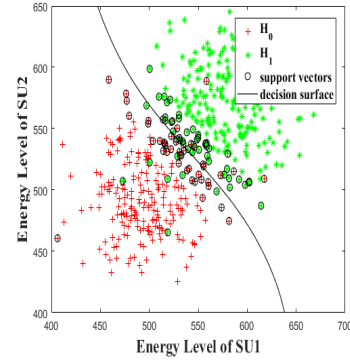


Fig. 3: Scatter plot of energy vectors classified by SVM when SNR=6dB

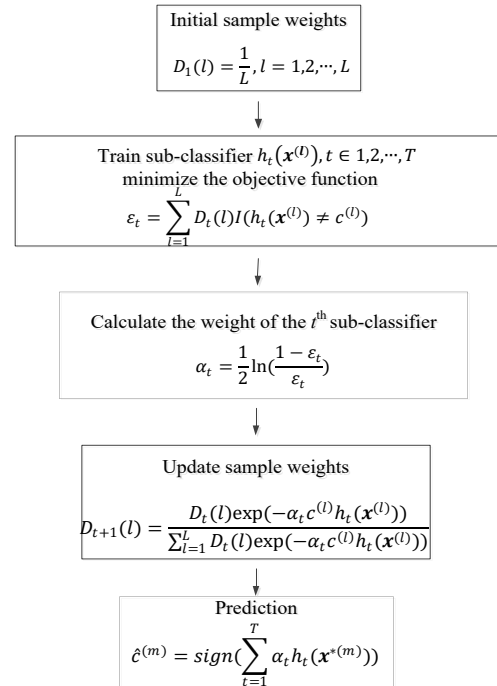


Fig. 4: AdaBoost classification mechanism.

drawbacks of poor classification of individual sub-classifiers [11][12]. The core idea is to train different sub-classifiers h_t on the same training samples. Decision stump, as one kind of weak classifier, is often used as sub-classifier in AdaBoost algorithm [13]-[15], and these weak sub-classifiers are grouped together to construct a final stronger classifier.

There are two kinds of weight distribution in AdaBoost. The first one is the weight of each training sample and the second one is the weight of each sub-classifier. The update of the weights of each training sample at each AdaBoost round ensures that correctly classified data receive lower weights and wrongly classified data receive higher weights in the next round. In this sense, the next sub-classifier is forced to focus on samples that have not been correctly classified in the previous round. It is up to the user to determine the structure of a sub-classifier that is employed to train the final stronger classifier.

As for the weights of each sub-classifier, the sub-classifier with larger weight indicates that it has better classification result than the sub-classifiers with lower weights. Predictive classification $h_t(\mathbf{x}^{*(m)})$ of the test samples is subsequently obtained by linearly combining these sub-classifiers based on their corresponding weights. In Fig. 4, the weight α_t is obtained from fitting a sub-classifier, h_t , by minimizing the objective function ε_t , which measures the discrepancy between the predictions and the true classes of the samples. In the next step, the weights of samples are updated. This algorithm can be used for classification by choosing suitable sub-classifiers.

In the first step, we need to assign the initial weight D_1 to each training sample as $1/L$. The SVM is adopted as the first sub-classifier (i.e., $h_t = h_{\text{SVM}}, t = 1$) and multiple decision stumps are used as the following sub-classifiers (i.e., $h_t = h_{\text{ds}}, t = 2, \dots, T$). After hybrid AdaBoost classification, we can get the weights of each sub-classifier as follows:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right) \quad (13)$$

where ε_t is the classification error at the t -th AdaBoost round by using the t -th sub-classifier which is defined as:

$$\varepsilon_t = \sum_{l=1}^L D_t(l) I(h_t(\mathbf{x}^{(l)}) \neq c^{(l)}) \quad (14)$$

where $h_t(\mathbf{x}^{(l)})$ is the predicted class of $\mathbf{x}^{(l)}$ by classifier h_t .

In hybrid AdaBoost algorithm, the correctly classified sample's weights increase and the falsely classified sample's weights decrease at the end of each AdaBoost round by equation:

$$D_{t+1}(l) = \frac{D_t(l) \exp(-\alpha_t c^{(l)} h_t(\mathbf{x}^{(l)}))}{Z_t} \quad (15)$$

where $D_t(l)$ denotes the l -th training sample's weight at the t -th AdaBoost round, and Z_t presents the normalization factor which is defined as:

$$Z_t = \sum_{l=1}^L D_t(l) \exp(-\alpha_t c^{(l)} h_t(\mathbf{x}^{(l)})) \quad (16)$$

Since the SVMs classification performance is generally better than that of the single decision stump, the weight of the SVM classifier α_1 is usually greater than that of the other single decision stump classifiers $\{\alpha_2, \alpha_3, \dots, \alpha_T\}$. Finally, these sub-classifiers are linearly combined to obtain a final classifier to predict the channel availability class on the basis of the test samples $\mathbf{x}^{*(m)}$. The classification result for the CSS is then obtained as:

$$\hat{c}^{(m)} = H(\mathbf{x}^{*(m)}) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}^{*(m)})\right) \quad (17)$$

Specifically, the proposed AdaBoost algorithm based on the hybrid structure of SVM and decision stumps is described in Algorithm 1.

Algorithm 1 SVM AND DECISION STUMPS BASED HYBRID ADABOOST CLASSIFICATION ALGORITHM

Input: $\{\bar{\mathbf{x}}, \bar{y}\}, \{\mathbf{x}^*, c^*\}, T$ (number of sub-classifiers).

Output: $\hat{c}^{(m)}$.

- 1: Initialize sample weight $D_1(l) = \frac{1}{L}$.
- 2: $\varepsilon_1 = \sum_{l=1}^L D_1(l) I(h_1(\mathbf{x}^{(l)}) \neq c^{(l)})$.
- 3: $\alpha_1 = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_1}{\varepsilon_1}\right)$.
- 4: $D_2(l) = \frac{D_1(l) \exp(-\alpha_1 c^{(l)} h_1(\mathbf{x}^{(l)}))}{Z_1}$.
- 5: **for** $t = 2$ **to** T **do**
- 6: $\varepsilon_t = \sum_{l=1}^L D_t(l) I(h_t(\mathbf{x}^{(l)}) \neq c^{(l)})$
- 7: $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$
- 8: $D_{t+1}(l) = \frac{D_t(l) \exp(-\alpha_t c^{(l)} h_t(\mathbf{x}^{(l)}))}{Z_t}$
- 9: **end for**
- 10: Predict the class of the m -th test sample

$$\hat{c}^{(m)} = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}^{*(m)})\right)$$

IV. SIMULATIONS RESULTS

In this section, the performance of the proposed SVM and decision stumps based AdaBoost classification algorithm is evaluated. We consider both a small scale CRN and a large scale CRN in the simulation. The layout of a small scale with 2 SUs participating in CSS is shown in Fig. 5, where are the SUs are located respectively at (-1,0) km and (0,0) km and the PU at (-0.6,-0.6) km. The large scale layout with 9 SUs is shown in Fig 6, where all the SUs are uniformly scattered within a 2.0 km 2.0 km square area and the single PU is located at (-0.6,-0.6) km.

The selected PU signal bandwidth ω is 5MHz, the sensing duration τ is 100 μ s, and the path-loss exponent α is 4. The shadow fading and the multi-path fading components are fixed as $\psi_n = 1$ and $v_n = 1$. In addition, the number of training samples L is 500 while the number of test samples is 500. The algorithms are implemented by using Matlab (R2016a) in a 64-bit computer with a core i5 processor (clock

speed of 3 GHz) and 8 GB RAM.

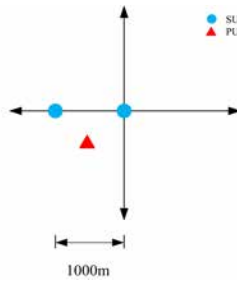


Fig. 5: Layout of the small scale CRN

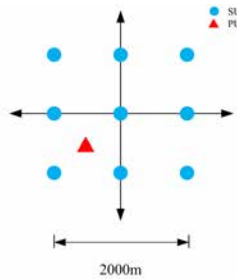


Fig. 6: Layout of the large scale CRN

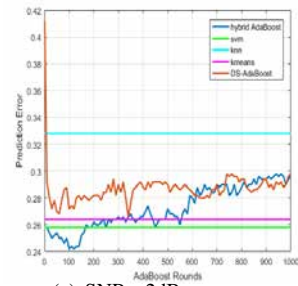
A. Prediction Error for Different Classifiers

The prediction errors of different machine learning techniques based CSS algorithms for different sets of cooperative SUs are evaluated and compared in Fig. 7 and Fig. 8, respectively. It is shown that the prediction error performance of the hybrid AdaBoost and the conventional AdaBoost algorithms deteriorate slightly when the number of AdaBoost rounds is getting large, whereas the prediction errors of SVM, K-Nearest Neighbor (KNN), and K-means remain consistent. The prediction error performance is even unacceptably deteriorating when the number of the cooperative SUs is as small as two. However, it is apparent that for the number of AdaBoost rounds being approximately 100, the proposed AdaBoost algorithm achieves the best performance among all the algorithms, regardless of the number of SUs participating in CSS.

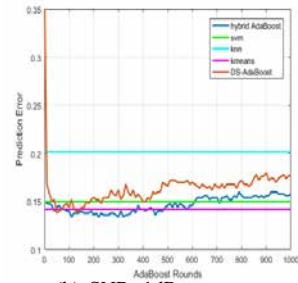
It's important to note that AdaBoost algorithm and the proposed hybrid AdaBoost algorithm both encounter the overfitting problem when the number of AdaBoost rounds is unnecessarily increased, which makes the prediction error increase. In particular, the overfitting point of both the AdaBoost and the hybrid AdaBoost are around 100 rounds. The hybrid AdaBoost algorithm achieves lowest prediction error 0.378, outperforming all the other algorithms under investigation.

B. Detection Probability for Different Classifiers

In addition to the prediction error, we also compare the detection probability performance of SVM, decision stumps

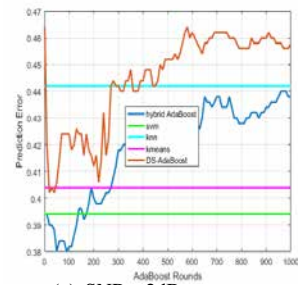


(a) SNR=-2dB

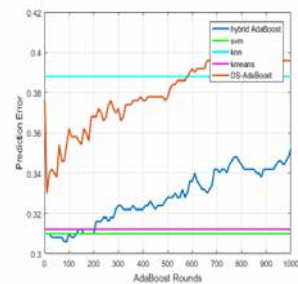


(b) SNR=1dB

Fig. 7: Prediction error when 9 SUs participate in CSS



(a) SNR=-2dB



(b) SNR=1dB

Fig. 8: Prediction error when 2 SUs participate in CSS

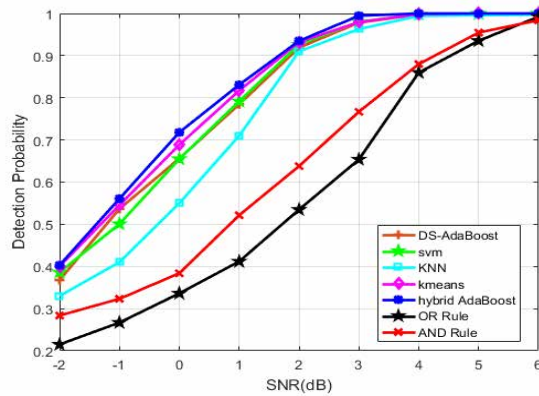


Fig. 9: The detection probability according to the SNR when the false alarm probability is 0.1 and there are 9 SUs.

based AdaBoost, the proposed hybrid AdaBoost, KNN, and K-Means. Hard decision fusion schemes such as AND and OR methods are evaluated for the purpose of comparison between the conventional CSS schemes and the machine learning based CSS algorithms, as shown in Fig. 9. The results are obtained for a desired false alarm probability of 0.1 when 9 SUs participate in CSS. This simulation results show that all the machine learning techniques based CSS schemes studied in this paper outperform the traditional AND and OR based CSS methods. Moreover, the proposed hybrid AdaBoost scheme performs best among all the CSS schemes.

V. CONCLUSION

In this paper, we propose a hybrid AdaBoost classification algorithm which uses SVM and decision stumps as sub-classifiers for cooperative spectrum sensing in cognitive radio networks. During spectrum sensing operation, the energy vectors collected from the cooperative SUs are used as the feature vectors in machine learning for determining the channel availability. The proposed hybrid AdaBoost algorithm achieves lower prediction error than the decision stumps based AdaBoost algorithm. Meanwhile, it exhibits the best performance in terms of detection probability in comparison of other machine learning algorithms and the traditional cooperative spectrum sensing methods. With the salient performance and technical merits, the proposed hybrid AdaBoost algorithm may serve as a practical solution for cooperative spectrum sensing.

REFERENCES

- [1] J. Mitola and G.Q. Maguire, "Cognitive Radio: Making Software Defined Radio More Personal," in *IEEE Personal Communications*, vol. 6, no. 4, 1999.
- [2] Axell, Erik, G. Leus, and E. G. Larsson, "Overview of spectrum sensing for cognitive radio," in *IEEE International Workshop on Cognitive Information Processing*, pp. 322-327, 2010.
- [3] Axell, E. Leus, G. Larsson, E.G. et al, "Spectrum Sensing for Cognitive Radio : State-of-the-Art and Recent Advances," *IEEE Signal Processing Magazine*, vol. 29, no. 3, pp. 101-116, 2012.
- [4] Mosleh, Somayeh, A. A. Tadaion, and M. Derakhshan, "Performance analysis of the Neyman-Pearson fusion center for spectrum sensing in a Cognitive Radio network," in *IEEE EUROCON*, pp. 1420-1425, 2009.

- [5] Guo Y, "Survey on spectrum sensing techniques in cognitive radio networks," in *Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, 2017.
- [6] Thilina K M, Choi K W, Saquib N, et al, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," in *IEEE Journal on selected areas in communications*, vol. 31, no. 11, pp. 2209-2221, 2013.
- [7] Pandey, Poonam, and R. Prabhakar, "An analysis of machine learning techniques (J48 & AdaBoost)-for classification," in *IEEE India International Conference on Information Processing*, pp. 1-6, March, 2017.
- [8] Freund, "Experiments with a new boosting algorithm," in *Thirteenth International Conference on Machine Learning Morgan Kaufmann Publishers Inc*, pp. 148-156, 1996.
- [9] S. Taylor and N. Cristianini, "Kernel Methods for Pattern Analysis," in *Cambridge University Press, Cambridge, UK*, 2004.
- [10] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning J*, vol. 20, no. 3, pp. 273-297, 1995.
- [11] Freund, "A short introduction to boosting," in *Journal-Japanese Society For Artificial Intelligence*, vol. 14, pp. 771-780, 1999.
- [12] Pandey P, Prabhakar R, "An analysis of machine learning techniques (J48 & AdaBoost)-for classification," in *IEEE India International Conference on Information Processing*, pp. 1-6, 2017.
- [13] Nicholas, "Using AdaBoost and Decision Stumps to Identify Spam E-mail," in *Natural Language Processing*, 2003.
- [14] Xie J, Lu Y, Lei Z, et al, "Boosting decision stumps to do pairwise classification," in *Electronics Letters*, vol. 50, no. 12, pp. 866-868, 2014.
- [15] Kawaguchi S, Nishii R, "Hyperspectral Image Classification by Bootstrap AdaBoost With Random Decision Stumps," in *IEEE Transactions on Geoscience & Remote Sensing*, vol. 45, no. 11, pp. 3845-3851, 2007.



SIJI CHEN received the B.S. degree from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2016. He is currently pursuing the M.S. degree in electronics and communication engineering with the Wireless Transmission Laboratory, School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. His main research interest is cognitive radios.